关于

本文所有资料均来自互联网

整理：谢志坚

QQ ：22922648

MSN ：xie_zhijian@msn.com

E-mail：xie_zhijian@163.com

http://xiezhijian.icpcn.com

入门教程目录

第一章

第一章 Linux基础知识

硬盘
硬盘是可以存储大量信息资源的媒介。我们平时看到的硬盘是方方正正的一块挺沉的铁匣子，但是其实硬盘是圆的，加上一些控制电路以后，为了便于携带而设计成为方形外壳。但是硬盘真是圆的，不骗你，要不怎么叫做盘呢？第二，你需要知道的是硬盘可以进行分区。可以把硬盘分区成为很多个分区，看来分区即是一个名次也是一个动词。什么叫做名次的分区呢？现在的硬盘越来越大，存放的资源越来越多。为了更合理的管理使用硬盘资源，更有效的提高查询速度和访问质量，我们可以把硬盘划分为很多分区，分别存放一些信息。当你需要读取某一个资料的时候，系统会自动使用最近的磁头去读取，加快读取速度。硬盘分区是从硬件的基础上对资源进行合理管理的手段。
分区已经了解了，但是我们不能只有分区啊，我们得用一个方法来控制分区。Windows操作系统使用了盘符这个概念，用单一的字母来表示硬盘上的不同分区，比如，C盘，D盘等等。在Windows2000以前，C盘对应了第一个分区，D盘对应第二个分区，依此类推。到了Winows2000，这个对应关系可以自己调整了。

在Linux里边，我们没有盘符这个概念，我们用目录的形式进行管理。
为了保证文章的完整性，简单说说目录吧，其实大家都很清楚的。如果我们把数以万计的水果扔在一个箩筐里面是不是很难找？但是如果我们把苹果放在叫做苹果的箩筐里面，把梨放在叫做梨的箩筐里面……这样是不是大大简化了我们寻找指定水果的速度？这些水果如果是我们的文件，那么我们用的箩筐就叫目录。新的问题出现了，即使把桃子扔在叫做桃子的箩筐中，我们仍然很难从好几千个桃子里面找出来自四川成都龙泉驿的桃子啊。聪明的你一定想到在桃子的箩筐中再放入很多叫做四川啊，河北啊，广州啊这些名字的箩筐，把不同产地的桃子放进去。对!就这样!我们在一个文件目录下面可以再建立新的文件目录，把文件进行非常细致的管理。到这里你明白了什么叫做目录，还明白了目录是可以嵌套的，这就可以了。目录技术是从软件的基础上对资源进行管理的方法。
回到Linux对于硬盘的管理里面来：在Linux里边，我们没有盘符这个概念，我们用目录的形式进行管理。把每一个分区和某一个目录对应，以后对这个目录的操作就是对这个分区的操作，这样就实现了硬件管理手段和软件目录管理手段的统一。这个把分区和目录对应的过程叫做挂载-mount。这种对应关系

可以由用户随时中断和改变。

SWAP交换分区
分区的知识大家已经熟习了，这里需要提一下交换分区。我们如果没有足够的内存，也许就不能运行某些大型的软件，解决的办法是在硬盘上划出一个区域来当作临时的内存，好像内存变大了。Windows操作系统把这个区域叫做虚拟内存，Linux把它叫做交换分区swap。虽然随着硬件价格的下降内存已经不大会限制我们了，但是交换分区这个概念却保留下来，而且有继续保留下去的原因，当然涉及到内核的问题，我们不多讨论，我们要说明的是，安装Linux时候建立交换分区时，如果你的内存只有64M或则128M，那么交换分区最好设置成为它的3倍，如果你的分区足够大，有256M或则512M，那么设置成为相同大小就可以了。但是请注意，最好大小为2的N次幂的数字。

分区格式
今天说了好多的与硬盘有关的问题，现在还要补充一个，就是分区格式。硬盘只是一个用电磁的方式纪录信息的媒介，我们可以通过不同的格式来管理。不同的操作系统选择了不同的格式，同一种操作系统也可能支持多种格式。微软的操作系统选择了fat16格式作为DOS的分区格式，往后windows95和windows98支持fat32格式，windowsNT使用NTFS4格式，windows2000以及XP支持NTFS5，也支持fat32，但是她们都不支持Linux上常见的分区格式。Linux是一个开放的操作系统，它最初使用ext2格式，后来使用ext3格式，但是它同时支持非常多的分区格式，包括很多大型机上UNIX使用的XFS格式，也包括微软的那些fat以及NTFS格式。
在给硬盘分区时就需要指定使用的分区格式。

掌握Linux的目录结构

既然目录是用来方便文件管理的工具，那么Linux下面不同的目录到底保存了哪一类的文件呢？

目录名称 意义

/bin 该目录中存放Linux的常用命令，在有的版本中是一些和根目录下相同的目录。
/boot 该目录下存放的都是系统启动时要用到的程序，当用lilo引导Linux时，会用到这里的一些信息
/dev 该目录包含了Linux系统中使用的所有外部设备，它实际上是访问这些外部设备的端口，你可以访问这些外部设备，与访问一个文件或一个目录没有区别。例如在系统中键入“cd /dev/cdrom”，就可以看到光驱中的文件；键入“cd /dev/mouse”即可看鼠标的相关文件。
/cdrom 该目录在刚安装系统时是空的，你可以将光驱文件系统挂在这个目录下， 例如“mount /dev/cdrom /cdrom”
/etc 该目录存放了系统管理时要用到的各种配置文件和子目录，例如网络配置文件、文件系统、X系统配置文件、设备配置信息、设置用户信息等。
/sbin 该目录用来存放系统管理员的系统管理程序。
/home 如果建立一个名为“xx”的用户，那么在/home目录下就有一个对应的“/home/xx”路径，用来存放该用户的主目录。
/lib 该目录用来存放系统动态连接共享库，几乎所有的应用程序都会用到该目录下的共享库
/lost+found 该目录在大多数情况下都是空的。但当突然停电、或者非正常关机后，有些文件就临时存放在这里。
/mnt 该目录在一般情况下也是空的，你可以临时将别的文件系统挂在该目录下。
/proc 可以在该目录下获取系统信息，这些信息是在内存中由系统自己产生的
/root 如果你是以超级用户的身份登录的，这个就是超级用户的主目录
/tmp 用来存放不同程序执行时产生的临时文件
/usr 用户的很多应用程序和文件都存放在该目录下

Linux支持的硬件
总的来说，Linux对于硬件的支持有三句话概括：越新的硬件越难支持，越小的厂家的产品越难支持，越来越多的硬件被支持!
所以推荐大家使用著名厂家的成熟产品。

下一章我们要介绍Linux的安装。我们选择的Linux发行版是Redhat Linux 7.3，在安装之前请大家一定熟习我们介绍的硬盘分区的知识。

上一页 目录 下一页

上一页 目录 下一页

第二章

第二章 Linux安装

下面我们介绍Linux的安装。各种Linux发行版本的安装各有不同，但是却大同小异。总的来说，除了国产的Linux以外，其他我们熟习的Linux都支持网络在线安装和硬盘安装。但是我们国内的用户使用宽带的现在并不多，硬盘安装方式又有一些麻烦，所以我们还是介绍光盘安装的方式。
Redhat Linux是国内市场上较为常见的Linux发行版本，甚至Redhat一度成为国人脑海中Linux的代名词。所以我选择Redhat作为第一次安装Linux时候的教材。其实MandrakeLinux安装界面比Redhat好看多了，不果似乎很难买到啊。往后的章节我还会详悉介绍的是Debian Linux的安装，不过那是后话了。

现在让我们假设你的硬盘是20G的，Windows98系统已经把这20G全部用完，具体如下：
C：前5G D：接下来5G
E：接下来5G F：最后5G
我们计划把Redhat安装在原来的F：盘上，那么现在请备份你的F：盘上的资料。

然后确定你的系统是由光盘引导的，如果不是，在电脑开机的时候按del键，进入主板BIOS设置，修改相关选项让你的电脑从光盘引导。然后把Redhat的第一章安装光盘放入光驱，开机。这时候你应该能够看到一些启动提示。相信大家都有大学英语的基本功，阅读这些东西是不成问题的。现在，你可以直接按回车键，启动图形界面的安装程序；也可以输入text然后回车，这样启动字符界面的安装。一般来说，图形界面占用更多的内存，安装程序会比字符界面下慢一些，但是图形界面更友好，考虑到这是我们第一次安装Linux，我们直接回车，进入图形界面的安装程序。

安装的第一步是要你确认它的版权信息。当然同意啦！Linux和Windows都有这一步，但是你所同意的版权协议却非常不同，具体的我们以后再说，总之你记住，Linux给你自由的世界，也许这个世界现在还有一些荒漠和戈壁；Windows却给你无限美好和宜人的牢笼。

现在请正确的选择你的键盘类型，一般pc104就可以了。好像在RedhatLinux下面还不支持win键。选择鼠标稍微麻烦一点，如果你的鼠标只有两个按钮并且没有滚轮，你一定要把那个“模拟三键鼠标”的选项

选中，因为在Linux下面，鼠标的第三键非常有用。

Redhat会讯问你是安装Workstation还是Server还是Custom类型的系统，或者选择update升级原有的Redhat。我们选择Custom！

现在配置硬盘参数，也就是分区，我们不要选择自动分区，那样不安全，非常不安全，特别不安全。选择手动分区吧，我会教你的。在继续之前请确保你熟习了我在上一章里面介绍的硬盘分区的基本知识，如果忘记了，快回去看看吧。注意：硬盘分区会清除掉你硬盘上相应分区以前保存的信息！

我相信你已经把F：盘上要备份的资料全部备份了。现在让我们来看看你的每一个Windows的盘符对应什么硬件设备。
一般来说，C盘对应/dev/hda1
D盘对应/dev/hda5
E盘对应/dev/hda6
F盘对应/dev/hda7
现在选中F盘对应的/dev/hda7，删除它。然后在空闲的硬盘空间（应该有5G，因为我们删除了一个5G的F盘）上建立一个新的分区，选择分区类型为SWAP，也就是交换分区。它的大小应该按照上一章的方法选择3倍于实际内存的大小或者适当的数字。但是请注意，最好大小为2的N次幂的数字。
添加完成这个交换分区的设定，现在我们再添加一个分区，这个分区类型选择为ext3，然后有一个mount point之类的选项，也就是我们说的挂装点，设置成“/”。把它的大小设置成为剩下的所有空间的大小。这样把整个硬盘空间都利用起来。
检查一下是不是还有什么没有作对？如果发现有问题，现在修改还来得及，因为我们所做的修改还没有保存到硬盘里面去。如果一切无误，选择下一步，现在硬盘的内容将按照我们设定的方式变化。
是不是有点激动？哈哈，说不定会搞砸呢？我以前第一次安装Redhat用了7个小时才搞懂什么叫做挂装点。整个硬盘从头到脚被分区无数次，安装完成以后什么资料都没有，哈哈，说不定你刚才的某一步做错了会和我一样而可怜呢。不过如果不是切肤之痛，怎会记忆犹新？搞Linux就是需要大胆的尝试，就象邓小平同志说的，实践是检验真理的唯一标准嘛！

看完这段文字，你的硬盘应该格式化完了，应该到了选择安装什么软件了。如果选择全部安装的话（在最下面有这个选项）你会得到一个巨大无比，巨慢无比的系统，大概2G吧。因为完全安装会安装所有的服务器程序个无关紧要的所有的东西。除非你知道自己在干什么，千万不要这么作。看看吧，如果你有打印机，就把打印支持选上。一般默认的选项足够了，但是我还是建议大家安装Software Development和Kernel Development两个选项，这样方便以后安装新的软件。窗口管理器有很多，一个KDE，一个Gnome，还有一些比较小巧的比如WindowMaker（这是我的最爱）。KDE的外观类似Windows，适合初学者上手，但是Gnome更漂亮，速度也更快。我很少使用KDE，也没有什么KDE的汉化经验，所以我就教大家使用Gnome吧。以后我还会教大家使用WindowMaker的。

继续下去似乎到了软件安装的过程，一般20到30分种吧，就可以安装完所有的软件。

这段时见不要休息，继续看我下面的内容！！！呵呵

好像这时候Redhat会要求选择启动的方式是使用Grub还是Lilo，也许你现在很郁闷，都不懂，那么我就告诉你，选择grub吧，因为……虽然我更熟习lilo，以后好讲给你听但是grub对于初学者来说不容易出错，安装grub的位置选择MBR就可以。只有这样才能保证你既能启动Linux，又能启动Windows。

我知道大家都很累了，所以我很快的说完下面的步骤：它会问你时区的问题，中国人当然选择亚洲的上海，没有北京的选项。如果你住在大西南，选择重庆也不错。下一步关键，如果你想得到正确的时间，

看清楚，它会问你是不是把本机时间和GMT时间对应，千万不要选啊！GMT实践是格林威治时间的意思。
继续下去到了配置显卡的地方，如果你的显卡是ATI的显卡或者常见的3dfx Banshee/Voodoo3+或者Matrox g200/g400或者SiS的话，那么恭喜你，你的显卡能够被轻易的识别出来。如果Redhat没有识别出你的显卡，也不要着急，跳过这一步，我们安装完成以后再来配置。
其实我的显卡是ATI的显卡，Redhat能够自动识别，但是我还是喜欢跳过去，安装完成以后再来搞它。
它还会问你启动时候是用图形界面启动还是字符界面启动，保险起见，选择字符界面！
下一步声卡也是一样的，如果你的声卡识别出来了，那么没什么好说的，没有的话，就跳过去，有办法的。
如果你的电脑有网卡，比如宽带或者ADSL，那么你还应该配置网络，但是网络的情况太多了，1000个人也许就有1000种情况，后面再说，跳过！

咦，我好像忘了介绍设置root密码的问题。什么是root啊？root就是超级用户，一定要细致的设定这个密码，因为一旦忘掉了，似乎只有重新安装整个Linux。其他的选项，我没有说到的，选择默认值就可以了。

好了，大功告成。

上一页 目录 下一页

上一页 目录 下一页

第三章

第三章 Linux 基本使用命令（基本篇）—（高级篇在后面会有，这一章从实用的角度出发介绍基本的操作）

现在请输入你的用户名和密码，当然，我们输入root，这样获得一切管理权限！

你一定非常希望立刻看到那些非常漂亮的图形界面，但是也许我要让你失望了。我建议在没有使用图形界面以前，首先熟练的掌握基本的Linux命令，这样才是一个真正的Linuxer。从哪里开始呢？

1. ls 列出文件和目录的命令

你一定很想知道你的电脑里面有哪些东西，现在执行命令ls，啊，怎么什么都没有？当然啦，这是你第一次登录到这个系统，你的默认位置是你的个人目录，而不是系统根目录。你还没有在这个目录里面存放任何的个人文件，当然什么都没有啦。如果你是用root用户登录的话，你的个人目录就是/root目录；如果你是用普通用户登录，比如叫做kris，那么kris的个人目录是/kris。前面的/是什么意思呢？就是“根”的意思，就是最前面的那个目录，在根目录下面建立有很多的子目录，我们在第一章已经讨论过了。

ls命令有很多的选项，常用的是：

-A 选项用来列出所有的文件，包括那些隐藏的文件。为什么我们要隐藏文件呢？道理和你为什么要把情书藏起来不让爸妈发现是一样的。就是为了保密啊。现在执行ls -A看看？是不是有一个隐藏文件“.bashrc”被显示出来啦？聪明的你一定奇怪的发现这个文件名前面有一个点，对！记住，只要文件名前面第一个字符是一个“.”，这个文件就是隐藏文件。一个目录名前面的第一个字符如果是“.”这个目录就是隐藏目录。
-l 这个选项用来显示一个列表，包含了这个目录下面所有的文件的绝大部分属性的列表。你可以每个文件的大小，所有者，你的权限还有修改日期等等。
-R R的意思就是recursive递归，明显这个选项让系统显示出这个目录下面的所有文件以外，还要显示出所有子目录下面的文件。也就是把我们那一大堆水果全部抖出来。
--color 这个选项特别有用，我估计大家的显示器都是彩显吧，什么？你的显示器还是黑白的？天哪！既然是彩显，那么我们可以让ls命令用不同的眼色代表不同的文件类型。比如可执行文件用绿色，普通文件是白色，目录是蓝色。也许你会问，目录也是文件吗？对的，在Linux里面一切都是文件，所有的硬件设备都用一个文件来代替，比如你的软驱，就是用/dev/fd0来代替的。目录也是一个文件。
--help 这个选项几乎是每一个Linux命令都有的，用来显示出该命令的帮助信息。

2. cd 和 mkdir 以及 rm 改变当然所在目录，建立新目录以及删除目录命令

趁热打铁的，刚才说了目录，我们每一次登录都有一个默认目录就是我们的个人用户目录。我们怎么才能到其他的目录去呢？cd就是用来改变当前所在的目录的。前面我们说过，“/”代表根目录，那么执行cd /就可以进入根目录。不试一下吗？
让我们看看根目录下面有哪些文件和子目录吧，执行ls，我们发现，根目录下面有一个目录名子特别变态，叫做usr，进去看看，cd usr，看看这里面有什么？你会发现一个更psycho（变态）的目录叫做src，进入src目录看看？没什么好玩的。那么我们现在回到刚才的usr目录，怎么做？是不是cd usr？执行试一下，好像不行，系统报告出错 cd: usr: No such file or directory。这是怎么搞的？问题在于我们现在所在的目录是/usr/src下，我们执行cd usr的意思是进入/usr/src/usr目录而不是/usr目录。正确的方法是cd /usr。
就好比你在中华美食的箩筐里面看到一个四川的箩筐，里面有一个成都的小箩筐，现在你进入以后发现成都的小箩筐里面有一种叫做“麻辣烫”的很辣的食品。你大饱口福以后想要吃一些甜点，于是准备去福州。你能站在成都的箩筐里面去福州吗？当然不行，福州并不在成都的箩筐里面啊，你应该进入“/中华美食/福州”而不是“/中华美食/四川/成都/福州”对不对？
好的，一个问题出现了，难道我每一次进入一个目录，都要用/usr/src...这么复杂的方式来表示吗？不一定。我们用“..”的方式来表示上一层目录。如果你现在在/usr/src目录下，进入/usr目录有两种办法：cd /usr和cd ..他们是一样的。

怎样才能知道我现在在哪个目录？用命令pwd，这个命令没有什么好说的，执行一次就知道了。

现在我想在我自己的个人目录里面建立一个目录叫做LoveLetter。我应该首先回到我自己的目录，这里有一个简单的方法，就是直接运行cd不带任何参数，这样就可以回到自己的目录，当然也可以cd /root或者cd /home/kris，看你是用什么用户登录的。
进入我自己的目录以后，建立新目录的命令是
mkdir 新目录名
我执行 mkdir LoveLetter 就可以建立一个新的叫做LoveLetter的目录。进入这个目录看看？什么都没有。不着急，慢慢来。我都不着急你急什么？
突然我想起这台电脑我的爸妈也要使用，他们看到我的情书目录怎么办？你忘了刚才我说的可以用加一个点“.”在前面的方法来隐藏目录和文件的？我们可以改变这个目录的名字，但是这个命令我准备等会儿讲，现在我们用一个很无聊的办法来完成这个要求。这个办法就是删掉刚才建立的oveLetter目录在新建一个.LoveLetter目录，之所以说这个办法很无聊，是因为我们现在是在做实验，如果来真的，你原意

删掉你的情书吗？是不是另有新欢啦？哈哈。

删除目录的命令其实也可以删除文件，就是rm。
rm 待删除的文件名/目录名
我记得Redhat会提示你是不是真的要删除。按y就是确定，按n就是取消。如果Redhat没有提示你，那么等会请根据我说的方法修改一下系统让它提示咱们。免得以后心痛。删除一个文件很简单。麻烦的是删除一个目录，如果一个目录里面已经有文件，rm是不让直接删除的，你必需先把目录里面的所有文件删除，再删除目录。但是有一个参数可以改变一下，就是 -rf ，这个参数有一定的危险性，因为即使系统本来要提醒一下是不是真的删除目录，加上这个参数也不会有提示了。执行rm 目录 -rf会在一眨眼的时间里面让你的资料下课！
那么我现在就删除LoveLetter目录了：rm LoveLetter -rf
建立一个新的目录mkdir .LoveLetter
现在ls看看，是不是看不到LoveLetter目录了？但是ls -A还是能看到的。所以这种隐藏方式只能偏偏自己，真正让你的文件安全的方式还是以后再讲吧。

3. mv 改变文件名和目录名的命令
cp 复制文件和目录命令
man 命令使用方法参考工具

mv 老文件名 新文件名
mv 老目录名 新目录名
就可以改变文件或者目录的名字。
我现在想要把刚才的这个目录.LoveLetter改名回去，因为这种无聊的隐藏方式很变态，我们有更高级的方法来做这样一件事情：就是不要告诉爸妈你的密码！！！
mv .LoveLetter LoveLetter

cp命令用来把一个文件复制成为一个新的文件，

cp 老文件名 新文件名

这个老文件明和新文件名如果在同一个目录下面，那么当然需要名字不一样，很简单的道理，如果文件名一样何必建立两个文件？如果新老文件在不同的目录，我们就可以让它们有相同的名子。下面的例子说明了这一点：

cp LoveLetter LoveLetter_yesterday 新的文件LoveLetter_yesterday和旧的LoveLetter在同一个目录，所以名子不一样。
cp LoveLetter /home/LoveLetter 新的文件在/home目录下面，但是旧的文件LoveLetter在某一个用户的个人目录下面，当然两者名子可以相同。

cp命令也可以复制整个目录，但是现在我们暂时不讲这么复杂。其实cp还有rm以及ls这些命令不仅是整个Linux的基本命令，更包含了非常多的功能。如果大家有兴趣，可以使用man

man 命令名字

比如man ls，这样就可以看到所有ls命令和参数的详悉解释，尤其是一部分常用的命令的man帮助已经由志愿者翻译了，大家看起来更容易。

一点幽默

好了，说了好多东西了，我想休息一下，给大家说一个有趣的事情，我们说了好多命令和目录的名子，你们是不是觉得有点奇怪。说实在话，我第一次看到usr这个目录时也不知道是什么意思，后来才发现以下对应关系：
usr -> user
ls -> list
mkdir -> make dir
rm -> remove
src -> source
mv -> move
cp -> copy
…
是不是很有趣，在UNIX世界，包括Linux世界，人们的想象力就是这么无敌！简写居然能简写成这样子。大家一般的想法是取一个单词的前三个或者前四个字母作为简写，可是UNIX的牛人就是喜欢把move简写成为mv，真不知道他们怎么想的。大家一起捉摸吧

4. nano 和 vi编辑文件的命令 和 cat 以及 more显示文本文件

nano是一个小巧自由，并且友好的编辑器，我认为nano更适合初学Linux的朋友使用。我们现在只学习怎样编辑一个文件以及怎样保存。

nano 文件名

如果你写的文件名已经存在，那么就打开并且编辑，否则就建立一个新的文件。编辑的方法还用说吗？呵呵，当你想要退出的时候，按ctrl+x，nano会问你是不是保存编辑的文件。按Y就是保存，按N就不保存。

nano最大好处在于用户可以不用记忆太多的操作键，大部分常用的功能的操作方法都在屏幕下放列出了。新手需要注意的是“^X”就是按住ctrl键不放再按X的意思。

下面简单的介绍vi。vi是一个非常强大的编辑软件。它太庞大了，足够写一本书专门来讲解。我们这里从使用的角度出发，讲一下vi的用法。
vi有两种模式，一种是命令模式，一种是编辑模式。进入vi以后，默认处于命令模式。

现在我们执行vi LoveLetter。进入以后，按一下键盘上的Insert功能键或者i键可以进入编辑状态，可以插入字符，再按一下Insert变成复盖模式，这两种模式的区别很容易体现，大家尝试一下就可以了。上下左右四个方向键可以移动光标。基本的编辑命令和Windows里面没有区别。是不是很容易呢？当你把需要的内容输入完成以后，我们要保存，这时候按一下ESC键从编辑模式回到命令模式，首先输入一个冒号“:”，也就是按住SHIFT键不放再按分号“;”这样首先输入一个“:”，然后，输入w，回车，就可以保存我们编辑的内容到LoveLetter文件。现在我们按一下Insert就可以继续编辑。再按ESC，输入“:”，再按w又可以保存。可是现在我们不需要保存，我们想要不保存就退出，怎么做呢？当我们输入w的时候是write的意思，保存，那么我们输入q就是quit退出的意思。好，输入q，回车，vi提示我们刚才进行的修改还没有保存，所以记住！一旦需要放弃我们的修改，不能直接用q命令退出，而需要用“q!”命令。输入q!，好了，退出了。
我们想看看我们刚才编辑的LoveLetter是不是真的保存好了，再vi LoveLetter，ok，看到了吧？现在我

们想要直接退出，就可以只输入“:q”就可以了，不用输入那个“!”因为我们没有修改文件内容。如果我们修改一下这篇文章，我们在退出的时候可以输入“ESC : wq”就可以了。不需要把w和q分成两次输入。

vi的最最基本用法说到这里差不多了，要是你还想多了解一些vi的知识，在进入vi以后直接按F1就可以了，有详悉的帮助和教学。

其实刚才我们想要看一下编辑的LoveLetter是不是保存好了，不用再vi进去的，只需要用命令

cat LoveLetter

就可以了。cat就是用来显示文本文件内容的命令。如果我们的文本文件很长，一个屏幕显示不完，cat是不会自动分页的。我们可以换用命令

more LoveLetter

more命令显示文本文件时，如果内容过多，会自动的在每一页结束时暂停下来，等到用户按一下空格键再继续。

5. 最重要的命令：halt reboot 关机和重新启动命令

在Linux里面，不能够直接用电源按钮关机，也不能直接用reset按钮重新启动，这对系统，尤其是硬盘有比较大的影响。关机命令是halt，重启动命令是reboot。其实还有shutdown命令完成类似功能，需要的话，请用今天学会的man命令学习使用。
<hr>

上一页 目录 下一页

上一页 目录 下一页

第四章

第四章 X的配置

虽然文章写得不好，但是我还是希望我的文章为公社的朋友提供帮助，和以前一样的，这个系列教程的文章我只在公社发表，希望大家捧场。

我的文章耽误了很久了，很抱歉。前段时间因为我一直忙着找工作，又生病一场，所以文章的事情一推再推，现在终于可以奉献给大家了。

这一章我主要介绍X系统用的主要配置文件XF86Config-4，我采用了对照的方法介绍，一边贴出我的XF86Config-4文件，一边介绍具体的内容。这篇文章对于大家没有什么立杆见影的帮助，不果可以让你对于X的只是有一个基本的了解。

XF86Config-4文件是X系统的主要配置文件。在Redhat 8以前版本中都叫做XF86Config-4这个名字，Redhat 8已经不再叫做这个名字。

编辑这个文件需要小心谨慎一点，因为一点错误，你的X将不能启动。不果没关系啦，改回来就是了学习Linux最好的办法当然还是求助于男人（man），大家有什么问题尽管看看man的帮助就是了。如果你要删除文件中的内容，最好不要直接的删除，而应该在前面加上#符号把它变成注释。

在Redhat 8以前的版本中，X的配置工具是Xconfigurator，在Debian中X的配种方法是：
dpkg-reconfigure xserver-xfree86
当然你都得用root的身份来运行。

在/usr/share/doc/xfree86-common/FAQ.gz文件中你可以看到具体的技巧。

第一段是Files段，这个部分用来配置X系统说能够使用的字体，每一行都代表一个目录，保存了具体的字体和字体的配置信息。
代码:

```
Section "Files"
FontPath "/usr/X11R6/lib/X11/fonts/xp"
FontPath "/usr/X11R6/lib/X11/fonts/XChinese"
FontPath "unix/:7100" # 这是本地字体服务器
# 如果本地字体服务器出了问题，我们可以使用下面的配置
FontPath "/usr/lib/X11/fonts/misc"
FontPath "/usr/lib/X11/fonts/cyrillic"
FontPath "/usr/lib/X11/fonts/100dpi/:unscaled"
FontPath "/usr/lib/X11/fonts/75dpi/:unscaled"
FontPath "/usr/lib/X11/fonts/Type1"
FontPath "/usr/lib/X11/fonts/Speedo"
FontPath "/usr/lib/X11/fonts/100dpi" #这两个字体是每一个X
FontPath "/usr/lib/X11/fonts/75dpi" #系统都必需安装的英文字体
EndSection
```

下面的是模块段，用来配置X系统加载的模块。
代码:

```
Section "Module"
Load "xtt" #gtk1使用的字体引擎，效果好，速度稍慢
Load "GLcore" #如果你是用的是Nvidia的显卡，似乎一定要注消掉这一行
Load "bitmap"
Load "dbe"
Load "ddc"
Load "dri"
Load "extmod"
# Load "freetype" #如果你使用了xtt模块，那么freetype模块就需要注消掉
Load "glx"
```

```
Load "int10"
Load "record"
Load "speedo"
Load "type1"
Load "vbe"
EndSection
```

下面的段是用来配置你的键盘的，属于“输入设备”
代码:

```
Section "InputDevice"
Identifier "Generic Keyboard" #这是你的键盘的名字，随便你啦:)
Driver "keyboard" #键盘的驱动…哇，键盘也有驱动
Option "CoreKeyboard" #如果你有多个键盘，那么你需要在这里指定哪一个键盘是主要的键盘
Option "XkbRules" "xfree86"
Option "XkbModel" "pc104" #键盘的分布格式，一般来说
Option "XkbLayout" "us" #美国104键盘是大家通用的。
EndSection
```

这里配置你的鼠标，当然你可以配置两个鼠标，如果你有的话
代码:

```
Section "InputDevice"
Identifier "Configured Mouse" #鼠标的名字
Driver "mouse" #鼠标的驱动
Option "CorePointer"
Option "Device" "/dev/input/mice"
#注意，这里很重要，这是鼠标的设备文件
#我的鼠标是光电鼠标，用的USB接口，对应的鼠标文件是/dev/input/mice
#如果你的鼠标是普通的滚轮鼠标，用的是PS2接口，那么你应该使用
#/dev/mouse或者/dev/psaux或者/dev/ttys0这个设备
Option "Protocol" "ImPS/2"
#这是鼠标的类型，如果不是是滚轮鼠标，那么使用PS/2
Option "Emulate3Buttons" "true"
#在Linux系统中，鼠标的第三个键非常有用，
#如果你的鼠标没有第三个键，那么我们应该允许使用双键同时点击来模拟
Option "ZAxisMapping" "4 5"
EndSection
```

下面的设备是显卡，这是最头痛的设备了，如果你的显卡太新潮，很有可能不能支持哦。Nvidia的GForce2显卡就必需自己编译显卡的驱动程序才能使用
代码:

```
Section "Device"
```

Identifier "Generic Video Card"
Driver "ati" #如果你是Nivida的显卡，这里应该是"nvidia"
EndSection

这个设备是显示器。
代码:

```
Section "Monitor"
Identifier "Generic Monitor" #显示器的名字
HorizSync 30-60 #显示器的频率，一半来说你的显示器
VertRefresh 50-75 #应该可以达到我的这个水平
#因为我的显示器是15"的老显示器了
#大家的电脑都比我的好吧？
Option "DPMS"
EndSection
```

下面是综合以上你的配置的设备的各种显示效果
代码:

```
Section "Screen"
Identifier "Default Screen" #效果的名字
Device "Generic Video Card" #你可以指定你的显卡的名字
Monitor "Generic Monitor" #指定你的显示器的名字
DefaultDepth 24 #默认的颜色深度
SubSection "Display"
Depth 1
Modes "1024x768"
EndSubSection
SubSection "Display"
Depth 4
Modes "1024x768"
EndSubSection
SubSection "Display"
Depth 8
Modes "1024x768"
EndSubSection
SubSection "Display"
Depth 16
Modes "1024x768" #在这里你可以指定扫描频率例如
#"1024x768 @ 85"就是用85mhz的频率
EndSubSection
SubSection "Display"
Depth 24
Modes "1024x768"
```

```
EndSubSection
EndSection
```

最终你必需定义下面的段用来告诉X服务器你使用的配置
代码:

```
Section "ServerLayout"
Identifier "Default Layout" #刚才我们给我们的配置取的名字
Screen "Default Screen" #给我们的效果取的名字
InputDevice "Generic Keyboard" #我们的键盘的名字
InputDevice "Configured Mouse" #我们的鼠标的名字
#这些名字一定要在前面的配置中已经定义
EndSection

Section "DRI"
Mode 0666
EndSection
```

一般来说我们X启动时候会遇到的问题是：
1：no screen found
这有可能是你没有正确的定义所需要的效果，也有可能是你的其他部分定义出错倒置你的效果不能实现
2：xtt和freetype的冲突，注消一个就可以了
3：驱动没有找到，如果你的显卡非常的新潮，那么多半是这个错误了，编译你的驱动吧…

上一页 目录 下一页

上一页 目录 下一页

第五章

第五章 Linux命令和技巧

1. mount umount 命令

在Linux里面想要使用你的软盘和光盘可不是像在Windows里面那么的危险。
想想看如果你在Windows里面安装一个光盘里面的软件或者玩游戏的时候，你的女朋友，甭管是第几个女朋友，因为对你每天打游戏而不做晚饭，乃至于别人做了晚饭你居然没时间吃，而气急败坏的按了一下光盘的Eject按钮……哈哈，害怕了？放心，在Linux里面这种事情绝对不用你操心。

要使用光盘，首先把光盘安放好以后，执行以下命令：

mount -t iso9660 /dev/cdrom /mnt/cdom 把光驱挂装到/mnt/cdrom目录

-t参数是指定光盘的文件系统类型，一般来说，光盘都是用的iso9660标准的格式。接下来的/dev/cdrom是你的光驱对应的设备文件，我们前面说过每一个硬件设备都对应了一个标准的设备文件。最后一个/mnt/cdrom目录是你的一个本地硬盘上的目录，在使用以前如果发现没有这个目录，可以建立一个。当然也不一定非要使用这个目录，你甚至可以很变态的使用mount /dev/cdrom /root命令把你的光驱挂装到你的的/root目录，这样一来你的/root的内容将不再是原来的内容，直到你取消挂载。

下面的命令取消挂载：

umount /mnt/cdrom 或 umount /dev/cdrom

这两个命令都可以把刚才挂装上去的光盘取消挂载，只有这样你的光盘才能取出。不行的话，你再执行一次挂装命令，然后按一下光驱上的Eject按钮，看看有没有什么反应啊？没有，对不对？这就是Linux的一个优点，虽然操作起来有点麻烦的感觉，但是想想看，非常安全和稍微的复杂，你选哪一个？

使用软盘的方法也很类似：

mount /dev/fd0 /mnt/floppy

/dev/fd0是你的软盘对应的设备文件，后面的自然是你的挂装目录。但是如果这个时候你想要尝试能不能在没有取消软盘的挂装就推出软盘，你就是一个非常BT的人了。有没有搞错？软盘的退出是机械的操作，你硬要按出来，我有什么办法？那么取消软盘挂载的方法也是很类似的：

umount /dev/fd0 或 umount /mnt/floppy

其实挂装时候第一个参数-t，可以省略，因为系统会自动的判断应该用什么样的格式来识别。不过知道了我们用的软盘或者光盘的格式，对于挂装一个ISO文件有帮助。

mount -t iso9660 -o loop /home/kris/somewhat.iso /mnt/cdrom

我们都知道现在网上有很多的ISO文件，就是把光盘的所有内容打包成为一个文件，方便网上的传送。我们获得这样的文件以后不一定非要用刻录机把它刻录成为光盘，我们只要用上面的命令就可以把光盘的ISO文件挂载到/mnt/cdrom目录了。第一个参数-t iso9660就是光盘使用的文件系统格式，第二个参数是loopback的意思。

取消挂装：umount /mnt/cdrom

2. 管道命令

管道命令是Linux系统的一个很重要的内容，这里我们只介绍一些简单的。

首先我们已经学过一个命令cat，用来显示一个文件的内容，但是你有没有遇到一个文件很长，一页显示不完的情况？如果你需要把一个很长的文件分页显示，每当你按一个键才显示下一页的话，你可以用more命令：

more LoveLetter

这样，LoveLetter的内容显示在屏幕上，没到一页就自动停下来，等待你的按键，然后才继续显示下面一页。

可是如果我们用ls命令显示一个目录的所有文件的时候遇到一页不能显示完的情况该怎么办？这就需要一个管道符号“|”，就是“\”按钮上面的字符:

```
ls /dev | more
```

下面，我们遇到一个新的问题，我们不仅仅想要看看/dev目录下面有哪些文件，我们还希望用一个文件来纪录刚才ls的命令显示的所有结果，难道需要用手工输入？不需要，我们介绍两个管道符号：“>”“>>”，也就是大于符号啦。

```
ls /dev > filenames.txt
```

还有一个管道符号是由两个大于号组成的“>>”，它和刚才“>”的区别在于一个大于符号的管道是用来创建一个新的文件filenames.txt，如果已经有了同名的文件就复盖掉以前的内容；而两个大于符号的管道是用来在已经存在的文件后面追加新的内容，如果没有这个文件就创建它。

现在我们vi filenames.txt看看一共有多少行，是不是1000多行啊？我们运行

```
ls /dev >> filenames.txt
```

再来vi filenames.txt看看，是不是一共2000多行啦？

其实这两个管道符号很容易记忆，向左边的箭头意思就是把前面命令的内容输入到后面的文件中去，刚才那个“|”管道符号是用来把前面命令的输出结果当作后面一个命令的输入数据。下面我们看一个和刚才“>>”刚刚相反的管道符号“<<”：

```
cat >> friends << "EOF"
> a
> b
> c
> EOF
```

运行第一行以后，意思就是把我们接下来的输入内容保存到friends文件里面，当我们输入“EOF”的时候结束。“<<”的意思就是向前面的命令传送一个数据。
中间我象征性的输入了a b c三行，每一行前面的“>”是系统自动产生的提示符号。第四行输入“EOF”系统就保存刚才输入的a b c三行到一个叫做friends的文件里面去。
现在我们cat friends看看是不是刚才的输入成功的保存了。怎么样，同样的命令，使用不同的管道符号，即可以显示文件内容，也可以用来创建文件，这就是管道符号的强大

3. Linux使用技巧（没有整理完成，有待大家的帮助）

Linux中有太多的使用技巧，这里慢慢的总结：
4. 基本命令总结

1） ls

ls --color=never *.txt > report 不显示文字颜色，将所有txt文件记录到report文件中

2) cd
cd ../ 到上一级目录
cd ../.. 到上二级目录
cd ~ 进入用户的个人目录，root用户就是/root目录，kris用户就是/home/kris目录

3） rm
rm -fr 目录名 直接删除一个目录和里面的所有目录，没有任何提示

4) cp
cp 源目录 目标位置 -R 复制原目录里面所有的文件和子目录的内容

5) echo
echo message “我爱你” 在字符界面下面显示一些内容，这里是“我爱你”

6) export
export LC_ALL= 定义一个变量LC_ALL并且设置为空NULL
export LANG=zh_CN.gb2312 定义一个变量LANG的值是zh_CN.gb2312
export http_proxy="http://xxx.xxx.xxx.xxx:port"
定义http代理服务器

7) find
find -iname [目录] 文件名 在指定目录里面（如果不指定目录就是当前目录）查找指定文件名的文件

grep
ps -A | grep some_command

用ps命令加上-A参数显示所有的当前运行的进程，用管道符号“|”加上grep命令检索是否有我们指定的some_command运行

9) lynx
lynx http://www.google.com/

字符界面的网络浏览器，我们这里用来浏览http://www.google.com

补充一点,没umount软盘就直接取出的话,可能软盘上更改的内容还没写进软盘,因为linux采用的是在系统空闲或缓冲区已满的时候才进行写盘操作的(提高效率??),有时你往软盘里拷贝了一个文件,没umount就直接取出软盘的话,可能那个文件还在缓冲区,没拷贝进去.因此,正确的方法应该是先umount /dev/fd0,然后等软盘的灯灭了后再取出软盘,这样就保证所有的写盘操作都已经完成.

系列教材一

目录

GNU通用公共许可证

GNU通用公共许可证 （1991. 6第二版）

序言

大多数软件许可证决意剥夺你的共享和修改软件的自由。对比之下，GNU通用公共许可证力图保证你的共享和修改自由软件的自由——保证自由软件对所有用户是自由的。GPL适用于大多数自由软件基金会的软件，以及由使用这些软件而承担义务的作者所开发的软件。(自由软件基金会的其他一些软件受GNU库通用许可证的保护)。你也可以将它用到你的程序中。

当我们谈到自由软件(free software)时，我们指的是自由而不是价格。我们的GNU通用公共许可证决意保证你有发布自由软件的自由(如果你愿意，你可以对此项服务收取一定的费用)；保证你能收到源程序或者在你需要时能得到它；保证你能修改软件或将它的一部分用于新的自由软件；而且还保证你知道你能做这些事情。为了保护你的权利，我们需要作出规定：禁止任何人不承认你的权利，或者要求你放弃这些权利。如果你修改了自由软件或者发布了软件的副本，这些规定就转化为你的责任。例如，如果你发布这样一个程序的副本，不管是收费的还是免费的，你必须将你具有的一切权利给予你的接受者；你必须保证他们能收到或得到源程序；并且将这些条款给他们看，使他们知道他们有这样的权利。

我们采取两项措施来保护你的权利。

(1)给软件以版权保护。

(2)给你提供许可证。它给你复制，发布和修改这些软件的法律许可。同样，为了保护每个作者和我们自己，我们需要清楚地让每个人明白，自由软件没有担保(no warranty)。如果由于其他某个人修改了软件，并继续加以传播。我们需要它的接受者明白：他们所得到

的并不是原来的自由软件。由其他人引人的任何问题，不应损害原作者的声誉。

最后，任何自由软件不断受到软件专利的威胁。我们希望避免这样的风险，自由软件的再发布者以个人名义获得专利许可证。事实上，将软件变为私有。为防止这一点，我们必须明确：任何专利必须以允许每个人自由使用为前提，否则就不准许有专利。

有关复制，发布和修改的条款和条件

0．此许可证适用于任何包含版权所有者声明的程序和其他作品，版权所有者在声明中明确说明程序和作品可以在GPL条款的约束下发布。下面提到的“程序”指的是任何这样的程序或作品。而“基于程序的作品”指的是程序或者任何受版权法约束的衍生作品。

也就是说包含程序或程序的一部分的作品。可以是原封不动的，或经过修改的和／或翻译成其他语言的(程序)。在下文中，翻译包含在修改的条款久每个许可证接受人(license)用你来称呼。

许可证条款不适用于复制，发布和修改以外的活动。这些活动超出这些条款的范围。运行程序的活动不受条款的限止。仅当程序的输出构成基于程序作品的内容时，这一条款才适用(如果只运行程序就无关)。是否普遍适用取决于程序具体用来做什么。

1．只要你在每一副本上明显和恰当地出版版权声明和不承担担保的声明，保持此许可证的声明和没有担保的声明完整无损，并和程序一起绘每个其他的程序接受者一份许可证的副本，你就可以用任何媒体复制和发布你收到的原始的程序的源代码。

你可以为转让副本的实际行动收取一定费用。你也有权选择提供担保以换取一定费用。

2．你可以修改程序的一个或几个副本或程序的任何部分，以此形成基于程序的作品。只要你同时满足下面的所有条件，你就可以按前面第一款的要求复制和发布这一经过修改的程序或作品。

a)你必须在修改的文件中附有明确的说明：你修改了这一文件及具体的修改日期。

b)你必须使你发布或出版的作品(它包含程序的全部或一部分，或包含由程序的全部或部分衍生的作品)允许第三方作为整体按许可证条款免费使用。

c)如果修改的程序在运行时以交互方式读取命令，你必须使它在开始进入常规的交互使用方式时打印或显示声明：包括适当的版权声明和没有担保的声明(或者你提供担保的声明)；用户可以按此许可证条款重新发布程序的说明；并告诉用户如何看到这一许可证的副本。(例外的情况：如果原始程序以交互方式工作，它并不打印这样的声明，你的基于程序的作品也就不用打印声明)。

这些要求适用于修改了的作品的整体。如果能够确定作品的一部分并非程序的衍生产品，可以合理地认为这部分是独立的，是不同的作品。当你将它作为独立作品发布时，它不受此许可证和它的条款的约束。但是当你将这部分作为基于程序的作品的一部分发布时，作为整体它将受到许可证条款约束。准予其他许可证持有人的使用范围扩大到整个产品。也就是每个部分，不管它是谁写的。

因此，本条款的意图不在于索取权利；或剥夺全部由你写成的作品的权利。而是履行权利来控制基于程序的集体作品或衍生作品的发布。

此外，将与程序无关的作品和该程序或基于程序的作品一起放在存贮体或发布媒体的同一卷上，并

不导致将其他作品置于此许可证的约束范围之内。

3．你可以以目标码或可执行形式复制或发布程序(或符合第2款的基于程序的作品)，只要你遵守前面的第1，2款，并同时满足下列3条中的1条。

a)在通常用作软件交换的媒体上，和目标码一起附有机器可读的完整的源码。这些源码的发布应符合上面第1，2款的要求。或者

b)在通常用作软件交换的媒体上，和目标码一起，附有给第三方提供相应的机器可读的源码的书面报价。有效期不少于3年，费用不超过实际完成源程序发布的实际成本。源码的发布应符合上面的第1，2款的要求。或者

c)和目标码一起，附有你收到的发布源码的报价信息。(这一条款只适用于非商业性发布，而且你只收到程序的目标码或可执行代码和按b)款要求提供的报价)。

作品的源码指的是对作品进行修改最优先择取的形式。对可执行的作品讲，完整的源码包括：所有模块的所有源程序，加上有关的接口的定义，加上控制可执行作品的安装和编译的script。作为特殊例外，发布的源码不必包含任何常规发布的供可执行代码在上面运行的操作系统的主要组成部分(如编译程序，内核等)。除非这些组成部分和可执行作品结合在一起。

如果采用提供对指定地点的访问和复制的方式发布可执行码或目标码，那么，提供对同一地点的访问和复制源码可以算作源码的发布，即使第三方不强求与目标码一起复制源码。

4．除非你明确按许可证提出的要求去做，否则你不能复制，修改，转发许可证和发布程序。任何试图用其他方式复制，修改，转发许可证和发布程序是无效的。而且将自动结束许可证赋予你的权利。然而，对那些从你那里按许可证条款得到副本和权利的人们，只要他们继续全面履行条款，许可证赋予他们的权利仍然有效。

5．你没有在许可证上签字，因而你没有必要一定接受这一许可证。然而，没有任何其他东西赋予你修改和发布程序及其衍生作品的权利。如果你不接受许可证，这些行为是法律禁止的。因此，如果你修改或发布程序(或任何基于程序的作品)，你就表明你接受这一许可证以及它的所有有关复制，发布和修改程序或基于程序的作品的条款和条件。

6．每当你重新发布程序(或任何基于程序的作品)时，接受者自动从原始许可证颁发者那里接到受这些条款和条件支配的复制，发布或修改程序的许可证。你不可以对接受者履行这里赋予他们的权利强加其他限制。你也没有强求第三方履行许可证条款的义务。

7．如果由于法院判决或违反专利的指控或任何其他原因(不限于专利问题)的结果，强加于你的条件(不管是法院判决，协议或其他)和许可证的条件有冲突。他们也不能用许可证条款为你开脱。

在你不能同时满足本许可证规定的义务及其他相关的义务时，作为结果，你可以根本不发布程序。例如，如果某一专利许可证不允许所有那些直接或间接从你那里接受副本的人们在不付专利费的情况下重新发布程序，唯一能同时满足两方面要求的办法是停止发布程序。

如果本条款的任何部分在特定的环境下无效或无法实施，就使用条款的其余部分。并将条款作为整体用于其他环境。

本条款的目的不在于引诱你侵犯专利或其他财产权的要求，或争论这种要求的有效性。本条款的主要目的在于保护自由软件发布系统的完整性。它是通过通用公共许可证的应用来实现的。许多人坚持应用这一系统，已经为通过这一系统发布大量自由软件作出慷慨的供献。作者 / 捐献者有权决定他 / 她是否通过任何其他系统发布软件。许可证待有人不能强制这种选择。

本节的目的在于明确说明许可证其余部分可能产生的结果。

8．如果由于专利或者由于有版权的接口问题使程序在某些国家的发布和使用受到限止，将此程序置于许可证约束下的原始版权拥有者可以增加限止发布地区的条款，将这些国家明确排除在外。并在这些国家以外的地区发布程序。在这种情况下，许可证包含的限止条款和许可证正文一样有效。

9．自由软件基金会可能随时出版通用公共许可证的修改版或新版。新版和当前的版本在原则上保持一致，但在提到新问题时或有关事项时，在细节上可能出现差别。

每一版本都有不同的版本号。如果程序指定适用于它的许可证版本号以及“任何更新的版本”。你有权选择遵循指定的版本或自由软件基金会以后出版的新版本，如果程序未指定许可证版本，你可选择自由软件基金会已经出版的任何版本。

10．如果你愿意将程序的一部分结合到其他自由程序中，而它们的发布条件不同。写信给作者，要求准予使用。如果是自由软件基金会加以版权保护的软件，写信给自由软件基金会。我们有时会作为例外的情况处理。我们的决定受两个主要目标的指导。这两个主要目标是：我们的自由软件的衍生作品继续保持自由状态。以及从整体上促进软件的共享和重复利用。

没有担保

11．由于程序准予免费使用，在适用法准许的范围内，对程序没有担保。除非另有书面说明，版权所有者和 / 或其他提供程序的人们“一样”
不提供任何类型的担保。不论是明确的，还是隐含的。包括但不限于隐含的适销和适合特定用途的保证。全部的风险，如程序的质量和性能问题都由你来承担。如果程序出现缺陷，你承担所有必要的服务，修复和改正的费用。

12．除非适用法或书面协议的要求，在任何情况下，任何版权所有者或任何按许可证条款修改和发布程序的人们都不对你的损失负有任何责任。包括由于使用或不能使用程序引起的任何一般的，特殊的，偶然发生的或重大的损失(包括但不限于数据的损失，或者数据变得不精确，或者你或第三方的持续的损失，或者程序不能和其他程序协调运行等)。即使版权所有者和其他人提到这种损失的可能性也不例外。

最后的条款和条件

如何将这些条款用到你的新程序

如果你开发了新程序，而且你需要它得到公众最大限度的利用。

要做到这一点的最好办法是将它变为自由软件。使得每个人都能在遵守条款的基础上对它进行修改和重新发布。为了做到这一点，绘程序附上下列声明。最安全的方式是将它放在每个源程序的开头，以便最有效地传递拒绝担保的信息。每个文件至少应有“版权所有”行以及在什么地方能看到声明全文的说明。

＜用一行空间给出程序的名称和它用来做什么的简单说明＞

版权所有(C)19xx(＜作者姓名＞

这一程序是自由软件，你可以遵照自由软件基金会出版的GNU通用公共许可证条款来修改和重新发布这一程序。或者用许可证的第二版，或者
(根据你的选择)用任何更新的版本。

发布这一程序的目的是希望它有用，但没有任何担保。甚至没有适合特定目的的隐含的担保。更详细的情况请参阅GNU通用公共许可证。

你应该已经和程序一起收到一份GNU通用公共许可证的副本。

如果还没有，写信给：

The Free Software Foundation，Inc，，675 Mass Ave，Cambridge， MA02139，USA还应加上如何和你保持联系的信息。

如果程序以交互方式进行工作，当它开始进人交互方式工作时，使它输出类似下面的简短声明11

Gnomovision第69版，版权所有(C)19XX，作者姓名，

Gnomovision绝对没有担保。要知道详细情况，请输人'show w'。

这是自由软件，欢迎你遵守一定的条件重新发布它，要知道详细情况，请输人'Show c'。

假设的命令‘shovr w’和‘show c’应显示通用公共许可证的相应条款。当然，你使用的命令名称可以不同于‘show w’和‘show
c，。根据你的程序的具体情况，也可以用菜单或鼠标选项来显示这些条款。

如果需要，你应该取得你的上司(如果你是程序员)或你的学校签署放弃程序版权的声明。下面只是一个例子，你应该改变相应的名称：

Ynyodyne公司以此方式放弃James Harker 所写的Gnomovision程序的全部版权利益。

＜Ty coon签名＞，1989. 4. 1

Ty coon付总裁

这一许可证不允许你将程序并人专用程序。如果你的程序是一个子程序库。你可能会认为用库的方式和专用应用程序连接更有用。如果这是你想做的事，使用GNU库通用公共许可证代替本许可证。

Linux安装的几点经验

我们一起在风雨中前行

Linux安装的几点经验

有关于Linux操作系统的好处许多文章都已经有介绍过了，想必大家一定都知道一些，前不久，我在我的电脑上安装了一套Linux操作系统（版本是Red Hat Linux 7.3），安装过程中有一些收获和体会，在此想通过这篇文章与各位交流一下，也算是体现一下Linux所倡导的共享精神。

1. 安装前一定要对硬盘规划好，即确定好Linux安装在哪个分区。因为Linux支持的分区格式：Linux Native（根分区）和Linux Swap（数据交换区），与Win 9x支持的分区格式FAT、Windows NT支持的分区格式NTFS，他们之间互不兼容。也就是说，如果你还要继续使用Win 9x、NT的话（相信大多数人会是这么做的），Linux就必须安装在单独的分区下。另外，Linux应安装在硬盘分区的最后一个扩展分区。例如：原来分区为C、D、E、F，一定要将Linux安装在F盘。如果你将Linux安装在了D盘，那么进入Win 9x后，原来的E盘成了D盘，F盘成了E盘。虽然各盘的软件都还能运行，但是桌面、开始菜单的快捷键却都已无效。更麻烦的是注册表内还是原先E、F盘的信息。

2. Linux的Swap分区swap的大小作的跟内存一样大就是了，最大128M。Native 分区的大小由你要安装的Linux组件多少决定，但最少要保证240MB。由于现在Linux的应用软件比较少，Native分区也不必留的太大。我建议Native分区不要超过2.5G。

3. 由于安装过程中会询问一些有关硬件的信息，因此要提前搜集好PC硬件方面的信息。硬件信息主要有显示器、显卡、鼠标、键盘等。特别是显示器的信息，将直接决定你安装Linux后，使用图形界面程序fvwm95的效果。

4. 如果Linux的安装事用Text方式的话那可比Windows的安装麻烦得多。首先，安装时它不支持鼠标，你必须频繁使用Tab、箭头、空格、回车等键做出选择。另外，如果你对满屏的英文没有十分把握，你手边最好放本字典。不过我们安装以默认的就可以，一路都事图形界面，它的安装和以前有了很大的改观。与windows的安装的区别在逐渐的缩小。

5. 由于Linux系统区分大小写，所以在安装、使用linux的过程中输入命令时，请注意大小写。

6. 下面谈谈Linux、Win 9x、NT多重启动方法。首先，在安装Linux的过程中，最好按照最后的要求制作好一张启动盘（Bootdisk）。还有安装过程中，系统会要你选择Linux的启动方法（我们一般选择默认grub）。这里有两种方法选择：Master Boot Record和First Sector of Boot Partition。第一种方法是直接从C盘，通过Linux系统的grub多重引导系统，选择某种操作系统启动；第二种方法是通过启动盘启动。这是因为毕竟现在大多数情况我们还是要使用Win 9x（或NT）系统。这样，平时启动时要进入到Windows系统（同时装有Win 9x、NT、MS－DOS时，还可通过NT的多重引导程序，做出进一步的选择，进入其中其他的一个系统），如果只安装一种linux我强烈建议使用第一种方法启动，在开机的时候可以通过选择菜单选择你所要进入的操作系统。如果选择第二中方法一般是安装两个以上的linux等特殊情况。然后再通过配grub（有时事lilo，不过他功能不如grub）多重引导。

Linux安装完成后，等待你的是深入学习它，了解它的好处。祝你在这个“新星”中，实现自己的梦

想。

linux文件系统

我们一起在风雨中前行

linux文件系统（hubertzou@linuxaid.com.cn）

Linux支持数量众多的本地文件系统类型和几种通过网络访问远程磁盘的文件系统，分析/proc/filesystems文件的内容即可找出当前Linux内核提供了对哪些文件系统的支持。

Linux默认直接支持的文件系统是ext2文件系统。

一，exe2文件系统

当前，ext2文件系统（即second extended filesystem）是Linux默认直接支持的文件系统。ext2文件系统和其他现代Unix使用的文件系统非常相似，但更接近于BSD系统所用的Berkeley Fast Filesystem。ext2文件系统的最大容量是4 TB，但目前它的最大文件大小被Linux内核限制为2 GB。

ext2的核心是两个内部数据结构，即superblock和inode。superblock是一个包含文件系统重要信息的表格，比如标签、大小、inode的数量等，它是对文件系统结构的基础性的、全局性的描述。因此，没有了superblock的文件系统将不再可用。由于这个原因，文件系统中不同位置存放着superblock的多个副本。

inode是基本的文件级数据结构，文件系统中的每一个文件都可以在其中一个inode中找到其描述。inode描述的文件信息包括：文件的创建和修改时间，文件大小，实际存放文件数据的块列表，等等。对于较大的文件，块列表可能包含附加数据块列表的磁盘位置（称为间接块），甚至有可能出现二重或三重的间接块列表。文件名字通过目录项（Directory Entry）关联到inode，目录项由“文件名字-inode”对构成。

二，创建和管理文件系统

Linux提供了大量用于创建和管理文件系统的工具。接下来我们将以新增磁盘（或分区）为线索，介绍各种要用到的工具。尽管完成这一任务的过程并不复杂，但这些操作还得在Linux命令行进行，这是因

为大多数图形界面的工具在这方面的支持非常有限。

1，确定设备文件

为新加入的物理磁盘（或磁盘分区）创建文件系统之前，首先必须知道该磁盘具体的设备名字。我们可以用好几种方法来找出它的设备名字。例如，可以通过dmesg命令分析启动日志（通过管道传送给grep，并搜索[hs]d[a-z]）；如果已经用syslog截取，则可以查看日志文件。另外，大多数Linux发行版都默认启用/proc文件系统，如果你也使用了它，则可以分析/proc/partitions文件，查看系统认可的所有磁盘分区的清单。

有时我们需要手工为磁盘创建设备文件，但这种情况极其罕见。设备文件可以用/dev目录下面的MAKEDEV脚本创建。例如，下面的命令将为第六个SCSI磁盘创建设备文件：

```
# cd /dev; MAKEDEV sdf
```

2、创建分区

确定磁盘的设备文件之后，接下来就要用fdisk之类的工具将磁盘分区。输入fdisk命令并在参数中指定目标磁盘

即可启动该工具：

```
# fdisk /dev/sdf
```

用fdisk修改好分区定义之后，可以用W命令保存更新后的分区表，用Q命令直接退出cfdisk则表示忽略所有已经作出的改动。只要对磁盘分区表作了任何实质性的改动（即，不仅仅指定分区标签或改变分区类型），则在继续处理之前

应当重新启动系统。

3，创建文件系统

用mkfs命令创建文件系统（或者，更精确地说是用mke2fs命令。mkfs是一个统一的前端工具，为所有不同文件系统类型的创建命令提供）。如下例所示，该命令最简单的形式是通过-t参数指定文件系统的类型，然后加上创建该文件系统的分区。下面这个命令在第二个SCSI磁盘的第二个分区上创建一个ext2文件系统：

```
#mkfs -t ext2 /dev/sdb2
# mke2fs /dev/sdb2
```

上面的第二个命令是直接使用mke2fs时的等价命令。

mke2fs的参数

-L label

指定分区的标签（最长11个字符）。

-b size

文件系统的块大小，以字节计，默认值1024。块是文件系统分配磁盘空间的单位。

-c

在创建文件系统之前检查分区的坏块。

-m percent

指定保留的文件系统空间百分比，这部分空间只能由root访问。默认值是磁盘的百分之五，这是传统Unix默认值的一半。在今天普遍使用大容量磁盘的情况下，即使是这个百分比也值得再加斟酌，比如可以将它降低到百分之二或者三。

-i ratio

指定每个inode的字节数，即为每个指定大小的字节块创建一个inode。默认值是4096，这个值所创建的

inode通常超过实际可能用到的，但一般没有必要改变它，因为节省下来的磁盘空间非常有限。另外，用“-N
数字”可以显式地指定创建inode的数量。
-v -q
分别表示详细输出或简略输出

4，其他工具

Linux下有许多用于检测现有ext2文件系统状态以及重新进行配置的工具。tune2fs就是其中一个，它的-l选项将列出指定ext2文件系统的各种属性。其中大多数属性的含义都非常明确，这些属性与可在mke2fs命令中指定的选项相对应。个别属性决定fsck何时对文件系统进行下一次检查。默认情况下，Linux将每隔20次系统启动或者30天对文件系统进行检查（这里的时间间隔周期以秒指定）。

这些文件系统属性中的很大一部分都可以用tune2fs的相应参数进行修改。例如，下面的命令通过设置两次检查
的时间间隔秒数（-i）为0禁止了根据时间周期执行的检查：

```
# tune2fs -i 0 /dev/hdb5
```

三，日志文件系统reiserfs

在1997年7月23日，Hans Reiser把他的基于平衡树结构的ReiserFS文件系统在网上公布 。 这是ReiserFS的第一次公开亮相。此后，ReiserFS一直在Hans
Reiser和领导下的开发小组下开发和发展，SuSE Linux也对它的发展起了重大的帮助。由于ReiserFS有一些很有用的特性，更主要的是它比ext2fs要快得多，所以它很快被很多人使用。目前，bluepoint linux 2.0 和suse linux 7.3都直接支持ReiserFS，更重要的是在Linux内核2.4.1版本开始已支持ReiserFS，估计ReiserFS很快就会普及。

ReiserFS是一种新型的文件系统，你可以选择它来装载你的Linux操作系统和其他应用程序和数据文件。它通过一种与众不同的方式--完全平衡树结构来容纳数据，包括文件数据，文件名以及日志支持。ReiserFS还以支持海量磁盘和磁盘阵列，并能在上面继续保很快的搜索速度和很高的效率。ReiserFS文件系统一直以来被用在高端Unix系统上如SGI等。

搜寻方式
ReiserFS是基于平衡树的文件系统结构，尤其对于大量文件的巨型文件系统，如服务器上的文件系统，搜索速度要比ext2快；ext2使用局部的二分查找法，综合性能比不上ReiserFS。

空间分配和利用情况

ReiserFS里的目录是完全动态分配的，因此不存在ext2中常见的无法回收巨型目录占用的磁盘空间的情况。ReiserFS里小文件（<4K）可以直接储进树，小文件读取和写入的速度更快，树内节点是按字节对齐的，小的文件可共享同一个硬盘块，节约大量空间。Ext2使用固定大小的块分配策略，也就是说，不到4K的小文件也要占据4K的空间，导致的空间浪费比较严重。

先进的日志机制

ReiserFS有先进的日志(Journaling/logging)机制，在系统意外崩溃的时候，未完成的文件操作不会影响到整个文件系统结构的完整性。

ext2虽然健壮性很强，但一旦文件系统被不正常地断开，在下一次启动时它将不得不进行 漫长的检查系统数据结构的完整性的过程，这是为了防止数据丢失而必需的操作。对于较大型的服务器文件系统，这种"文件系统检查"可能要持续好几个小时，在很多场合下这样长的时间是无法接受的，解决这个问题一种技术"日志文件系统"。在日志的帮助下，每个对数据结构的改变都被记录下来，日志在机制保证了在每个实际数据修改之前，相应的日志已经写入硬盘。正因为如此，在系统突然崩溃时，在下次启动几秒钟后就能恢复成一个完整的系统，系统也就能很快的使用了。

支持海量磁盘和优秀的综合性能

ReiserFS是一个相当现代化的文件系统，相比之下，ext2虽然性能已经很好了，但其设计还只是19世纪80年代的水准。ReiserFS的出现，使Linux拥有了像Irix/AIX那样的高档商用Unix才有的高级文件系统。ReiserFS可轻松管理上百G的文件系统，在企业级应用中有其用武之地，由于它的高效存储和快速小文件I/O特点，它在桌面系统上也表现出色：启动X窗口系统的时间ReiserFS比ext2少1/3。而ext2则无法管理2G以上的单个文件，这也使得ReiserFS在某些大型企业级应用中比ext2要出色。

ReiserFS一个最受人批评的缺点是每升级一个版本，都将要将磁盘重新格式化一次，这个缺点也正在改进中。
http://www.devlinux.com/namesys
下载后用root身份登录进入系统，切换至/usr/src/linux目录，执行命令：
#gunzip /path/to/linux-2.2.16-reiserfs-3.5.22-patch.gz
#patch -p1 -i /path/to/linux-2.2.16-reiserfs-3.5.22-patch
在做完上面的两步后，重新编译kernel，将reiserfs编译进kernel或做成一个模块。
Reiserfs的相关应用程序在 /usr/src/linux/fs/reiserfs/utils 目录下。你可以用下面方法安装它们：

#mkdir bin
#make
#make install
最后，你可以用“fdisk”命令重新分区或用“mkreiserfs”命令重新格式化一个已经存在的分区。指定reiserfs类型加载这个分区，如“mount
-t reiserfs /dev/hda2/download”，这样你就可以使用这个新的分区了。

Linux系统安装前的准备工作

我们一起在风雨中前行

Linux系统安装前的准备工作

在您开始安装Linux之前,您必须做以下的准备工作:

1. 备份您硬盘上的所有重要数据! 首先建议您将您硬盘上的重要数据备份到软盘/光盘/磁带上,以免发生意外时,给您造成不必要的损失.

2.如果采用硬盘,网络安装,或者bios不支持光盘启动，则要准备一，两张安装软盘。

所需的img文件可在redhat安装光盘的/images/目录下找到,所需的img文件（/images/boot.img，bootnet.img)

制作方法一，在dos下：

把光碟上的/dosutils/rawrite.exe 拷贝到dos系统下，准备一张刚dos格式化的软盘。

"C:

awrite ***.img"

制作方法二，在unix(linux)下：

准备一张刚dos格式化的软盘,不要mount上系统。

"dd if=***.img of=/dev/fd0"

3. 准备Linux分区

Linux拥有自己的文件系统 (Linuxext2)，所以，你在安装linux时它应该拥有自己特别的分区类型。这意味着你不能将Linux安装到一个已经存在的
DOS或Windows98分区上。所以您必须将硬盘上的分区进行调整，同时要在硬盘上留有足够的而且未被分配的磁盘空间来安装Linux，要达到这样的目的您可以通过下面的途径来做:

A,备份硬盘上的数据，通过dos下的fdisk对硬盘进行分区（这个办法一般不推荐使用，除非你使用的事一个崭新的硬盘。或者对硬盘上的所有数据进行北方，不然所有的数据都将会丢失）

B,先在widows下将最后的一个分区的数据往前几个分区移动使其空出来（对于重要的数据请备份，以防不测）。如果你对linux下的分区软件Disk
Druid了解的话，你就可以直接进行安装。要你是一个新手的话，我建议你去找dos版的Patition
magic，这是一个非常专业而且功能强大
的分区编辑软件，您可以使用该工具对您的硬盘分区进行重新划分。不过现在你只要在dos下将刚才整理出来的空间的格式转换为linux et2的格式，那样你在等下安装linux的时候在分区的那项的时候你就可以按照默认进行。

4.收集你的机器的所有硬件的信息，这些硬件大概包括如下几种 主板（主板上的芯片主型号） 声卡 显卡 鼠标 键盘 网卡等，这些信息一般你都可以在windows中找到。当然还有你的机器的说明书，这个可不要忘了。

Linux与其他操作系统的区别

我们一起在风雨中前行

Linux与其他操作系统的区别

Linux可以与MS-DOS、OS/2、Windows等其他操作系统共存于同一台机器上。因为它们均为操作系统，具有一些共性，但是互相之间各有特色，有所区别。

目前运行在PC机上的操作系统主要有Microsoft的MS-DOS、Windows、Windows NT、IBM的OS/2等。早期的PC机用户普遍使用MS-DOS，因为这种操作系统对机器的硬件配置要求不高，而随着计算机硬件技术的飞速发展，硬件设备价格越来越低，人们可以相对容易地提高计算机的硬件配置，于是开始使用Windows、Windows
NT等具有图形界面的操作系统。Linux是新近被人们所关注的操作系统，它正在逐渐为PC机的用户所接受。那么，Linux与其他操作系统的主要区别是什么呢？下面从两个方面加以论述。

首先看一下Linux与MS－DOS之间的区别.在同一系统上运行Linux和MS-DOS已很普遍，就发挥处理器功能来说，MS-DOS没有完全实现x86处理器的功能，而Linux完全在处理器保护模式下运行，并且开发了处理器的所有特性。Linux可以直接访问计算机内的所有可用内存，提供完整的Unix接口。而MS-DOS只支持部分Unix的接口。

就使用费用而言，Linux和MS-DOS是两种完全不同的实体。与其他商业操作系统相比，MS-DOS价格比较便宜，而且在PC机用户中有很大的占有率，任何其他PC机操作系统都很难达到MS-DOS的普及程度，因为其他操作系统的费用对大多数PC机用户来说都是一个不小的负担。Linux是免费的，用户可以从internet上或者其他途径获得它的版本，而且可以任意使用，不用考虑费用问题。

就操作系统的功能来说，MS-DOS是单任务的操作系统，一旦用户运行了一个MS-DOS的应用程序，它就独占了系统的资源，用户不可能再同时运行其他应用程序。而Linux是多任务的操作系统，用户可以同时运行多个应用程序。

再看一下Linux与OS/2、Windows、Windows NT之间的区别。

从发展的背景看，Linux与其他操作系统的区别是，Linux是从一个比较成熟的操作系统发展而来的，而其他操作系统，如Windows
NT等，都是自成体系，无对应的相依托的操作系统。这一区别使得Linux的用户能大大地从Unix团体贡献中获利。因为Unix是世界上使用最普遍、发展最成熟的操作系统之一，它是七十年代中期发展起来的微机和巨型机的多任务系统，虽然有时接口比较混乱，并缺少相对集中的标准，但还是发展壮大成为了最广泛使用的操作系统之一。无论是Unix的作者还是Unix的用户，都认为只有Unix才是一个真正的操作系统，许多计算机系统（从个人计算机到超级计算机）都存在Unix版本，Unix的用户可以从很多方面得到支持和帮助。因此，Linux做为Unix的一个克隆，同样会得到相应的支持和帮助，直接拥有Unix在用户中建立的牢固的地位。

从使用费用上看，Linux与其他操作系统的区别在于Linux是一种开放、免费的操作系统，而其他操作系统都是封闭的系统，需要有偿使用。这一区别使得我们能够不用花钱就能得到很多Linux的版本以及为其开发的应用软件。当我们访问Internet时，会发现几乎所有可用的自由软件都能够运行在Linux系统上。有来自很多软件商的多种Unix实现，Unix的开发、发展商以开放系统的方式推动其标准化，但却没有一个公司来控制这种设计。因此，任何一个软件商（或开拓者）都能在某种Unix实现中实现这些标准。OS/2和Windows

NT等操作系统是具有版权的产品，其接口和设计均由某一公司控制，而且只有这些公司才有权实现其设计，它们是在封闭的环境下发展的。

unix系统发展史

我们一起在风雨中前行

unix系统发展史
一、Multics计划

1965年，AT&T贝尔电话实验室、通用电气公司、麻省理工学院MAC课题组一起联合开发一个称为Multics的新操作系统。Multics系统的目标是要向大的用户团体提供对计算机的同时访问，支持强大的计算数据存储，以及允许用户在需要的时候容易地共享他们的数据。

1969年Multics的原始版本在GE645计算机上运行了，但它既没能提供预定的综合服务，而且，连它自己也不清楚究竟什么时刻算达到开发目标。在这种情况下，AT&T贝尔电话实验室首先退出了这一项目。

二、UNIX系统的萌芽

Multics项目中止后，AT&T贝尔电话实验室中参加开发的主要人员开始寻找其他有趣的项目。这其中，KenThompson写了一个名叫“太空漫游”的游戏程序。他找到了一台基本无人使用的DEC
PDP-7计算机运行他的程序。然而由于DEC PDP-7缺少程序开发程序，以致于Thompson不得不在另一台运行GECOS操作系统的Honeywell635上交叉汇编子他的程序，然后再通过纸带加载到DEC
PDP-7上。
为了方便太空漫游的开发，Thompson和Dennis Ritchie一起着手开发DEC PDP-7上的操作环境：
1.一个简单的文件系统，它后来深化成称之为System V文件系统的早期版；
2.一个进程子系统和shell（命令解释器）。
这样这个系统就足以满足他们的需要，他们将这个操作环境命名为UNIX。
Thompson和Ritchie后来开发了B语言（后来Ritchie又将其发展为C语言），并扩展了UNIX系统，UNIX系统便在AT&T贝尔电话实验室流行起来。

1973年11月，他们用C语言彻底重写，这是UNIX操作系统迈向成功之路的关键一步。完成后，Thompson和Ritchie共同在《ACM通信》上发表了首篇UNIX论文，The
UNIX Time Sharing System。

三、UNIX系统的发展

而由于AT&T公司受到美国反托拉斯法的诉讼，与联邦政府签署了一份协议，这份协议不允许AT&T公司生产与电话和电报服务无关的设备，也不允许从事除“公用通信服务”外的任何商业活支。这样，AT&T公司就陷入了不能经营计算机产品的尴尬局面。

而自从在《ACM通信》中发表了关于UNIX的论文后，越来越多的人希望能够获得UNIX软件和源码。为了不违背与政府签署的协议，AT&T公司既不出售UNIX、也不为其作广告、不对其提供技术支持。而同时为了满足大家的需要，AT&T公司在签署简单协议的前提下，将UNIX系统无偿地提供给大学，以供教学与研究。

由于AT&T公司对UNIX的策略，使得UNIX在全世界范围内快速发展。到1975年底，它已经传播到西至耶路撒冷的Hebrew大学，南至澳大利亚的新南威尔士大学，北至加拿大多伦多大学的世界各地。

1979年1月发行的UNIX版本是一个真正可移植的UNIX系统，它对其后的UNIX发展有着深远的影响。该版本最初是运行在PDP-11和Interdata
8/32上的，该系统更加健壮，而且提供了比版本6更强大的功能，但其相当慢。许多领取了许可证的用户针对这一情况从不同方面对系统的性能进行了改进，AT&T又将这些改进中的许多加入了其以后的UNIX发行中。

这种使用者与开发者之间的合作精神正是UNIX快速成长和不断发展的关键因素。不幸的是，当UNIX在商业上取得很大成功之后，这种精神就不存在了。

四、UNIX系统最重要的分支----BSD UNIX

1974年12月，加州大学伯克利分校获得了UNIX许可证。在其后的几年里，包括Bill Joy和Chuck Haley在内的一组研究生为UNIX开发了几个实用工具，其中包括ex编辑器---vi的前身和一个Pascal编译器。他们将这些软件绑定成一个称为“伯克利软件销售（BSD）”的软件包，并在1978年春以每个许可证50美元出售。最初的BSD发行仅包括应用程序和实用工个，并没有对操作系统进行修改和再发行。Joy早期的贡献之一就是C
Shell。

1978年伯克利得到一台VAX-11/780以及由新泽西州Holmdel的BTL小组移植的UNIX/32V。VAX-11采用32位体系结构，可以有4GB的地址窨，而物理内存仅为2MB。与此同时，Ozalp
Babaoglu为VAX设计了一个分页式的虚存系统，并将其加入到UNIX系统中。这就是1979年末发行的3BSD，是由伯克利发行的第一个操作系统。

伯克利在虚存方面的工作促使美国国防部高级研究项目署（DARPA）决定投资支持伯克利的UNIX系统开发。DARPA项目的目的之一就是将TCP/IP网络协议族集成到UNIX系统中。在DARPA的资助下，伯克利又推出了几个BSD发行，统称4BSD。

五、纯UNIX系统----System V

AT&T公司与美国司法部的法律大战终天在1982年达到终点，一部里程碑式的法律出台了。根据该法令，地区性运营公司都从AT&T公司中分离出来，成立一系列的“贝尔宝宝”，
贝尔电话实验室也独立出来，改名为贝尔实验室。同时，AT&T也被重新允许进入计算机市场。
贝尔实验室先后在1982年发行了System III，1983年发行了System，1984年发行了System V Release 2（SVR2），1987年发行了SVR3。
System V UNIX也引入了许多新特征，新设施。最有代表性的就是进程间通信机制。
六、UNIX系统的商业化

UNIX系统的不断发展吸引了许多计算机公司。它们急急忙忙地开始将UNIX商业化，出售自己的UNIX版本。每一家都是以AT&T或BSD为蓝本，将它们移植到自己的硬件上，并加上一些自己的“增值”功能。

1982年，Bill Joy离开了伯克利，与人合资成立了Sun公司，基于BSD开发、发行了SunOS，后来又基于SVR4发行了Solaris。

微软与SCO合作开发发行的XENIX。
此后，SCO公司将SVR3移植到386上，并将其命名为SCO UNIX。
还有一此大型的硬件公司，配合自己的计算机系统，也开发出一许多不同的UNIX版本：
IBM公司开发的AIX操作系统----运行在RS 6000小型机上；
HP公司开发的HP-UX----运行在HP工作站上；
Digital公司开发的Digital UNIX。
还有著名教授Tanenbaum为了教学开发的MINIX系统；
以及我们可爱的小企鹅----LINUX操作系统。
每种商业变体都增加了许多新特征，这些功能中的许多后来又被加入到主线系统中，这样逐步推动UNIX系统向前发展。
七、UNIX系统标准
UNIX系统变体的泛滥造成了许多不兼容问题，尽管所有的变体从某种程度“看起来都像UNIX”，但它们在许多重要方面却千差万别。最初，业界根据AT&T的System
V发行版、伯克利的BSD发行版的不同一分为二。然而，后来随着商业变体不断涌现，情况越来越糟。
Sys unix系统发展史
LinuxAid 工程师 fjxufeng
一、Multics计划

1965年，AT&T贝尔电话实验室、通用电气公司、麻省理工学院MAC课题组一起联合开发一个称为Multics的新操作系统。Multics系统的目标是要向大的用户团体提供对计算机的同时访问，支持强大的计算数据存储，以及允许用户在需要的时候容易地共享他们的数据。

1969年Multics的原始版本在GE645计算机上运行了，但它既没能提供预定的综合服务，而且，连它自己也不清楚究竟什么时刻算达到开发目标。在这种情况下，AT&T贝尔电话实验室首先退出了这一项目。

二、UNIX系统的萌芽

Multics项目中止后，AT&T贝尔电话实验室中参加开发的主要人员开始寻找其他有趣的项目。这其中，KenThompson写了一个名叫“太空漫游”的游戏程序。他找到了一台基本无人使用的DEC
PDP-7计算机运行他的程序。然而由于DEC PDP-7缺少程序开发程序，以致于Thompson不得不在另一台运行GECOS操作系统的Honeywell635上交叉汇编子他的程序，然后再通过纸带加载到DEC
PDP-7上。
为了方便太空漫游的开发，Thompson和Dennis Ritchie一起着手开发DEC PDP-7上的操作环境：
1.一个简单的文件系统，它后来深化成称之为System V文件系统的早期版；
2.一个进程子系统和shell（命令解释器）。
这样这个系统就足以满足他们的需要，他们将这个操作环境命名为UNIX。
Thompson和Ritchie后来开发了B语言（后来Ritchie又将其发展为C语言），并扩展了UNIX系统，UNIX系统便在AT&T贝尔电话实验室流行起来。

1973年11月，他们用C语言彻底重写，这是UNIX操作系统迈向成功之路的关键一步。完成后，Thompson和Ritchie共同在《ACM通信》上发表了首篇UNIX论文，The
UNIX Time Sharing System。
三、UNIX系统的发展

而由于AT&T公司受到美国反托拉斯法的诉讼，与联邦政府签署了一份协议，这份协议不允许AT&T公司生产与电话和电报服务无关的设备，也不允许从事除“公用通信服务”外的任何商业活支。这样，AT&T公司就陷入了不能经营计算机产品的尴尬局面。

而自从在《ACM通信》中发表了关于UNIX的论文后，越来越多的人希望能够获得UNIX软件和源码。为了不违背与政府签署的协议，AT&T公司既不出售UNIX、也不为其作广告、不对其提供技术支持。而同时为了满足大家的需要，AT&T公司在签署简单协议的前提下，将UNIX系统无偿地提供给大学，以供教学与研究。

由于AT&T公司对UNIX的策略，使得UNIX在全世界范围内快速发展。到1975年底，它已经传播到西至耶路撒冷的Hebrew大学，南至澳大利亚的新南威尔士大学，北至加拿大多伦多大学的世界各地。

1979年1月发行的UNIX版本是一个真正可移植的UNIX系统，它对其后的UNIX发展有着深远的影响。该版本最初是运行在PDP-11和Interdata

8/32上的，该系统更加健壮，而且提供了比版本6更强大的功能，但其相当慢。许多领取了许可证的用户针对这一情况从不同方面对系统的性能进行了改进，AT&T又将这些改进中的许多加入了其以后的UNIX发行中。

这种使用者与开发者之间的合作精神正是UNIX快速成长和不断发展的关键因素。不幸的是，当UNIX在商业上取得很大成功之后，这种精神就不存在了。

四、UNIX系统最重要的分支----BSD UNIX

1974年12月，加州大学伯克利分校获得了UNIX许可证。在其后的几年里，包括Bill Joy和Chuck Haley在内的一组研究生为UNIX开发了几个实用工具，其中包括ex编辑器---vi的前身和一个Pascal编译器。他们将这些软件绑定成一个称为“伯克利软件销售（BSD）”的软件包，并在1978年春以每个许可证50美元出售。最初的BSD发行仅包括应用程序和实用工个，并没有对操作系统进行修改和再发行。Joy早期的贡献之一就是C
Shell。

1978年伯克利得到一台VAX-11/780以及由新泽西州Holmdel的BTL小组移植的UNIX/32V。VAX-11采用32位体系结构，可以有4GB的地址窨，而物理内存仅为2MB。与此同时，Ozalp
Babaoglu为VAX设计了一个分页式的虚存系统，并将其加入到UNIX系统中。这就是1979年末发行的3BSD，是由伯克利发行的第一个操作系统。

伯克利在虚存方面的工作促使美国国防部高级研究项目署（DARPA）决定投资支持伯克利的UNIX系统开发。DARPA项目的目的之一就是将TCP/IP网络协议族集成到UNIX系统中。在DARPA的资助下，伯克利又推出了几个BSD发行，统称4BSD。

五、纯UNIX系统----System V

AT&T公司与美国司法部的法律大战终天在1982年达到终点，一部里程碑式的法律出台了。根据该法令，地区性运营公司都从AT&T公司中分离出来，成立一系列的“贝尔宝宝”，
贝尔电话实验室也独立出来，改名为贝尔实验室。同时，AT&T也被重新允许进入计算机市场。
贝尔实验室先后在1982年发行了System III，1983年发行了System，1984年发行了System V Release
2（SVR2），1987年发行了SVR3。
System V UNIX也引入了许多新特征，新设施。最有代表性的就是进程间通信机制。
六、UNIX系统的商业化

UNIX系统的不断发展吸引了许多计算机公司。它们急急忙忙地开始将UNIX商业化，出售自己的UNIX版本。每一家都是以AT&T或BSD为蓝本，将它们移植到自己的硬件上，并加上一些自己的“增值”功能。

1982年，Bill Joy离开了伯克利，与人合资成立了Sun公司，基于BSD开发、发行了SunOS，后来又基于SVR4发行了Solaris。

微软与SCO合作开发发行的XENIX。
此后，SCO公司将SVR3移植到386上，并将其命名为SCO UNIX。
还有一此大型的硬件公司，配合自己的计算机系统，也开发出一许多不同的UNIX版本：
IBM公司开发的AIX操作系统----运行在RS 6000小型机上；
HP公司开发的HP-UX----运行在HP工作站上；
Digital公司开发的Digital UNIX。
还有著名教授Tanenbaum为了教学开发的MINIX系统；
以及我们可爱的小企鹅----LINUX操作系统。
每种商业变体都增加了许多新特征，这些功能中的许多后来又被加入到主线系统中，这样逐步推动UNIX系统向前发展。
七、UNIX系统标准
UNIX系统变体的泛滥造成了许多不兼容问题，尽管所有的变体从某种程度“看起来都像UNIX”，但它们在许多重要方面却千差万别。最初，业界根据AT&T的System
V发行版、伯克利的BSD发行版的不同一分为二。然而，后来随着商业变体不断涌现，情况越来越糟。
System V与BSD系统的不同，影响到了用户编程。此时，你已不可能编写一个复杂应用不需任何修改变可以在两种系统上运行。
接着，各种商业版本的掺合，使得开发应用的程序员不得不花费大量的精力来保证它们的程序可以运行于各种不同的UNIX系统。
在这种局面下，人们强烈要求出台一组标准接口，许多小组开始了这方面的工作。其中被广泛认可的有三个标准：
1.AT&T的System V接口定义----SVID；
2.IEEE POSIX（可移植操作系统标准）规范；

操作系统简述

我们一起在风雨中前行

操作系统简述

（LinuxAid 工程师 fjxufeng）

一、计算机软件与操作系统

功能强大的计算机自从走进了人类的生活就发挥着越来越重要的作用，随着时代的发展，计算机已与人们的日常生活息息相关。不能不说计算机软件日新月异的发展在此起着关键作用。可以这么说，离开了软件，计算机就成了废铜烂铁。计算机机软件大致可以分为两类：系统软件和应用软件。系统软件

用于管理计算机资源，并为应用软件提供一个统一的平台。应用软件则在系统软件的基础上实现用户所需要的功能。
而操作系统(Operation System)则是最基本的系统软件，它控制计算机的所有资源关提供应用程序开发的基础。

二、操作系统诞生的原因

计算机是由CPU、内存、磁盘、显卡、声卡等许许多多设备组成的，而且这些设备的厂商众多，品种繁多，而且不同厂商生产的同种设备虽然完成同种功能，但是具体细节却存在千差万别。为了正确地管理和使用这些设备来实现具体的应用，这样程序员就得了解和掌握各种设备的工作原理。
而且对于同种设备，由于不同的硬件厂商在实现细节上的差异使得程序员再次陷入了复杂的硬件控制的深渊！必须找到一种方法使得程序员从苦海中脱离出来！多年的研究与发展终于使得这个愿望成为现实。这个解决方法就是在硬件的基础上加载一层软件来管理整个系统。这个软件通过设备驱动程序来与计算机硬件打交道，通过一系列的功能模块将整个计算机硬件系统抽象成为一个公共、统一、开放的接口—虚拟机，从而使得程序员不必再陷入各种硬件系统的具体细节！这一层软件就是操作系统。

三、什么是操作系统

操作系统是一个大型的软件系统，其功能复杂，体系庞大。从不同的角度看的结果也不同，正是“横看成岭侧成峰”，下面我们通过最典型的两个角度来分析一下。

1.从程序员的角度看

正如前面所说的，如果没有操作系统，程序员在开发软件的时候就必须陷入复杂的硬件实现细节。程序员并不想涉足这个可怕的领域，而且大量的精力花费在这个重复的、没有创造性的工作上也使得程序员无法集中精力放在更具有创造性的程序设计工作中去。程序员需要的是一种简单的，高度抽象的可以与之打交道的设备。将硬件细节与程序员隔离开来，这当然就是操作系统。从这个角度看，操作系统的作用是为用户提供一台等价的扩展机器，也称虚拟机，它比底层硬件更容易编程。

2.从使用者的角度看

从使用者的角度来看，操作系统则用来管理一个复杂系统的各个部分。操作系统负责在相互竞争的程序之间有序地控制对CPU、内存及其它I/O接口设备的分配。

比如说，假设在一台计算机上运行的三个程序试图同时在同一台打印机上输出计算结果。那么头几行可能是程序1的输出，下几行是程序2的输出，然后又是程序3的输出等等。最终结果将是一团糟。这时，操作系统采用将打印输出送到磁盘上的缓冲区的方法就可以避免这种混乱。在一个程序结束后，操作系统可以将暂存在磁盘上的文件送到打印机输出。从这种角度来看，操作系统则是系统的资源管理者。

四、操作系统发展历史

下面我们结合计算机的发展历史来回顾一下操作系统的发展历程。

1.第一代计算机（1945-1955）：真空管和插件板

40年代中期，美国哈佛大学、普林斯顿高等研究院、宾夕法尼亚大学的一些人使用数万个真空管，构建了世界上第一台电子计算机。开启计算机发展的历史。这个时期的机器需要一个小组专门设计、制造、编程、操作、维护每台机器。程序设计使用机器语言，通过插板上的硬连线来控制其基本功能。
这个时候处于计算机发展的最初阶段，连程序设计语言都还没有出现，操作系统更是闻所未闻！

2.第二代计算机（1955-1965）：晶体管和批处理系统

这个时期计算机越来越可靠，已从研究院中走出来，走进了商业应用。但这个时期的计算机主要完成各种科学计算，需要专门的操作人员维护，并且需要针对每次的计算任务进行编程。

第二代计算机主要用于科学与工程计算。使用FORTRAN与汇编语言编写程序。在后期出现了操作系统的雏形：FMS（FORTRAN监控系统）和IBMSYS（IBM为7094机配备的操作系统）

3.第三代计算机（1965-1980）：集成电路芯片和多道程序

60年代初，计算机厂商根据不同的应用分成了两个计算机系列，一个针对科学计算，一个针对商业应用。随着计算机应用的深入，对统一两种应用的计算机需求出现了。这时IBM公司试图通过引入System/360来解决这个问题。与这个计划配套，IBM公司组织了OS/360操作系统的开发，然后复杂的需求，以及当时软件工程水平低下使得OS/360的开发工作陷入了历史以来最可怕的“软件开发泥潭”，诞生了最著名的失败论著----《神秘的人月》。虽然这个开发计划失败了，但是这个愿望却成为了计算机厂商的目标。此时，MIT、Bell Lab（贝尔实验室）和通用电气公司决定开发一种“公用计算机服务系统”----MULTICS，希望其能同时支持数百名分时用户的一种机器。结果这个计划的研制难度超出了所有人的预料，最后这个系统也以失败结束。不过，MULTICS的思想却为后来的操作系统很多提示。60年代末，一位贝尔实验室曾参加过MULTICS研制工作的计算机科学家Ken Thompson，在一台无人使用的PDP-7机器上开发出了一套简化的、单用户版的MULTICS。后来导致了UNIX操作系统的诞生。目前UNIX操作系统主导了小型机、工作站以及其他市场。也是至今最有影响力的操作系统之一，而Linux也是UNIX系统的一种衍生，下一讲我们将专门介绍一下UNIX的发展历史。

4.第四代计算机（1980-今）：个人计算机

随着计算机技术的不断更新与发展，计算机神奇般地闯入了人们的生活，现在以低廉的价格就可以获得强大计算能力的计算机。价格不再是阻拦计算机普及的门槛时，降低计算机的易用性就显得十分重要！由于UNIX系统的本身特点，使得其不太适合于在运行在个人计算机上，这时就需要一种新的操作系统。在这一历史关键时候，IBM公司由于低估了PC机的市场，并未使用最大的力量角逐这一市场，这时Intel公司趁机进入，成为了当今微处理器的老大。同时善于抓住时机的微软公司的总裁比尔·盖茨适时地进入了这一领域，用购买来的CP/M摇身一变成为MS-DOS，并凭借其成为个人计算机操作系统领域的霸主。虽然是苹果公司在GUI方面先拔头筹，但由于苹果公司的不兼容、不开放的市场策略，未能扩大战果，这时微软又适时地进入了GUI方面，凭借WINDOWS系统再次称雄！

五、操作系统构成

一般来说，操作系统由以下几个部分组成：

1）进程调度子系统：进程调度子系统决定哪个进程使用CPU，对进程进行调度、管理。

2）进程间通信子系统： 负责各个进程之间的通信。

3）内存管理子系统：负责管理计算机内存。

4）设备管理子系统：负责管理各种计算机外设，主要由设备驱动程序构成。

5）文件子系统：负责管理磁盘上的各种文件、目录！

6）网络子系统：负责处理各种与网络有关的东西。

六、操作系统结构设计

操作系统有多种实现方法与设计思路，下面仅选取最有代表性的三种做一简单的叙述。

1.整体式系统结构设计 这是最常用的一种组织方式，它常被誉为“大杂烩”，也可说，整体式系统结构就是“无结构”。

这种结构方式下，开发人员为了构造最终的目标操作系统程序，首先将一些独立的过程，或包含过程的文件进行编译，然后用链接程序将它们链接成为一个单独的目标程序。

Linux操作系统就是采用整体式的系统结构设计。但其在此基础上增加了一些形如动态模块加载等方法来提高整体的灵活性，弥补整体式系统结构设计的不足。

2.层次式系统结构设计 这种方式则是对系统进行严格的分层，使得整个系统层次分明，等级森严！这种系统学术味道较浓！实际完全按照这种结构进行设计的操作系统不多，也没有广泛的应用。

可以这么说，现在的操作系统设计是在整体式系统结构与层次式系统结构设计中寻求平衡。

3.微内核系统结构设计 而微内核系统结构设计则是近几年来出现的一种新的设计理念，最有代表性的操作系统有Mach和QNX。微内核系统，顾名思义就是系统内核很小！比如说QNX的微内核只负责：

¨ 进程间的通信
¨ 低层的网络通信
¨ 进程调度
¨ 第一级中断处理

七、操作系统横向比较

计算机历史中出现了许许多多的操作系统，然后大浪淘沙，无情地淘汰了许多，只留下一些经历过市场考验的：

1.桌面操作系统：

1）MSDOS：Intel x86系列的PC机上的最早的操作系统，微软公司产品，曾经统治了这个领域，现在已逐渐被自家兄弟WINDOWS
9x系列所代替，现在除了一些低档机外已不多见。

2）Windows 9x：微软公司产品，从Windows 3.x发展而来，现在是基于Intel
x86系列的PC机上的主要操作系统，也是现然个人电脑中装机量最大的操作系统。面向桌面、面向个人用户。

3）Mac OS：苹果公司所有，界面友好，性能优异，但由于只能运行在苹果公司自己的电脑上而发展有限。但由于苹果电脑独特的市场定位，现在仍存活良好。

2.服务器操作系统：

1）UNIX系列：UNIX可以说是源远流长，是一个真正稳健、实用、强大的操作系统，但是由于众多厂商在其基础上开发了有自己特色的UNIX版本，所以影响了整体。在国外，UNIX系统可谓独树一帜，广泛应用于科研、学校、金融等关键领域。但由于中国的计算机发展较为落后，UNIX系统的应用水平与国外相比有一定的滞后。

2）Windows NT系列：微软公司产品，其利用Windows的友好的用户界面的优势打进服务器操作系统市场。但其在整体性能、效率、稳定性上都与UNIX有一定差距，所以现在主要应用于中小企业市场。

3）Novell Netware系列：Novell公司产品，其以极适合于中小网络而著称，在中国的证券行业市场占有率极高，而且其产品特点鲜明，仍然是服务器系统软件中的长青树。

4）Linux操作系统

常见linux介绍

常见linux介绍

目前我们所能接触到的linux主要有Red Hat、Slackware、Debian、SuSE、OpenLinux、TurboLinux、Red
Flag、Mandarke、
BluePoint等。

（1）Red Hat http://www.redhat.com

以容易安装著称，初学者安装这个版本，遇到挫折的机会几乎是零，如果您对安装Win9x已驾轻就熟的话，Red Hat Linux的安装一定难不倒您。

Red Hat另一个优点是它的RPM（Red Hat Package Manager）；以往在安装软件时，最让使用者伤脑筋的是：软件在解开压缩前先要新建一个目录，然后将软件搬进去解压，解压后，有些部份可能需要搬到另一个目录中去，当要搬移的项目多时，做这些工作就是件苦差事了。而RPM「包裹管理者」就针对这一点，能将所有要安装的路径全部安排好，当使用者解开有「.rpm」扩展名的文件时，会将当初打包该文件时设定好的路径档案先检查一次，然后依照档案里的设定，将各个文件解开，送到它们应该去的地方；不只如此，它还会制作安装记录，当使用者要移除其中任一个rpm文件时，系统会根据安装记录将该文件反安装，这种做法绝对准确，不会像Windows那样会移除不该拿掉的东西。

Red Hat Linux可以说是相当成功的一个产品，Red Hat公司有「官方版本」（official）供使用者购买，也提供了自由的FTP站供大众直接下载，官方版本与自由下载版本差异在于，官方版本多提供了一些商用软件和印刷精美的说明书。

（2）Slackware http://www.cdrom.com

这是个老字号的门派了，前几年玩Linux的人，几乎都用这套系统。它可完全「手工打造」个人需求的特性，让很多目前已是高手级的玩家仍念念不忘，Slackware在国内用得很多，也许用来做服务器，性能会好些。最新版本安装的过程已改善了不少，各位想要「完全掌控」情况的朋友可以一试。

（3）Debian http://www.debian.org

您通常会在Debian字眼后看见GNU Linux的字样，该派别目前是大家公认的结构最严谨、组织发展最整齐的，它也有一个包裹管理系统称之为「dpk」（Debian
Package），所作的事情和Red Hat的「rpm」异曲同工，使整体文件的管理更加方便。Debian的原始程序代码都是遵循GNU的方式开放的，所以它完全符合开放原代码精神，不像其它的Linux都或多或少的保留了一部份程序代码不开放（Red
Hat是直到6.0版才全部开放的）。最新版本有一部份程序设计师，将Debian移植到其它平台上，所以也可以在其它平台上看见其踪影。

（4）SuSE http://www.suse.com

这是一套在欧洲相当受欢迎的版本，它和XFree86合作开发x86上 的X Server。SuSE安装时可以选择显示德文或英文，它还有自己的一套设定程序叫做「SaX」，可以让使用者较方便的设定，它的安装套件也采用RPM模式，所以要安装、升级与移除程序都非常方便，目前版本8.0。

（5）OpenLinux http://www.caldera.com

这是由Caldera公司推出的版本，并不是很「Open」的一个版本，网上可以下载其Lite版本，但正式版本是要money的，因为整个套件中有许多商用软件，所以并没有提供网络下载的服务。

（6）TurboLinux http://www.pacific.com

由Pacific HiTech公司发展的套件，该套件在日本市场占有一席之地，从安装到使用接口都是日文的，在国内它与清华大学及研究机构合作研发了中文版本，在国内造成了一股Linux潮流，目前已推出7.0的简体版本，8.0的日文版本。

（7）Red Flag http://www.redflag-linux.com

这是由中科红旗软件技术有限公司推出的中文版本的linux，该linux在众多的中国linux用户中占有一定的比例。可以从网络上下载其红旗桌面版。目前桌面版的最高版本为3.2。同时红旗针对服务器市场，专门推出了红旗服务器版本，其目前最高版本为3.0。

（8）Mandarke Linux

它的吉祥物是一个黑色的魔术帽，它其实参照了Redhat的基础上制成的，它继承了许多Redhat的优点，还加上了许多迎合Linux初学者的功能，如美丽的图形化安装界面。7.0版本开始走向成熟，赢得了不少用户。而且现在自从推出8.0
8.1以及8.2版后，使得Mandarke也获得了较高的用户占有率。刚从Windows中走出来的朋友，可以一试。挺不错的。

（9）BluePoint Linux

我觉得是这做得最成功的一款中文Linux发行版，说心里话，蓝点还是挺有创新，挺挖得深的一个Linux厂商。但是其稳定性不是太好，我觉得适于桌面，不适于做务器。不过蓝点不再出新版本了，目前的版本为BluePoint
2.0

其他还有中软linux等等，由于篇幅以及资料收集方面的限制就不再一一介绍。请大家见谅。

2005-1-20 11:27:25 上一页 首页 下一页

2005-1-20 上一页 首页 下一页

关于开放源代码的定义

我们一起在风雨中前行

关于开放源代码的定义

开放源代码并不仅仅意味着对源代码的访问权。开放源代码软件的发布条款必须满足以下条件：

1. 自由地再发布

如果被发布的软件是由不同来源的程序组成的，许可证不得限制任何当事人或组织（party）销售或赠送作为被发布软件成分之一的开放源码软件。
许可证不得从此项销售中索取使用费或其它任何费用。（理由）

2. 源代码

程序必须包括源代码，必须允许以源代码方式发布、还必须允许以编译后的形式发布。如果产品的某个部分没有与源代码一同发布，那么必须提供通行的、不需要支付合理范围之外的任何费用的手段以获得源代码---从网络上免费下载是一种可取的方式。源代码必须是程序员对其进行修改的最佳形式。故意地使源代码变得含混晦涩是不允许的。也不允许给出预处理器或翻译器处理的中间结果。（理由）

3. 派生作品

许可证必须允许修改软件和派生软件，并且必须允许它们按照原软件的许可证的条款进行发布。（理由）

4. 作者的源代码的完整性

只有在许可证允许与源代码一同发布"补丁文件"（该"补丁文件"以在创建时对程序进行修改为目的）时，许可证才能限制对修改形式的源代码的发布。许可证必须明确地允许发布由修改后的源代码生

成的程序。许可证可以要求派生的作品采用不同的名称或不同的版本号以区别于原来的软件。（理由）

5.　不得歧视任何个人或团体

许可证不得歧视任何个人或者由多人组成的团体。（理由）

6.　不得歧视任何应用领域（fields of endeavor）

许可证不得限制任何人把程序应用于任何领域。例如，不得规定程序不能应用于商业领域或基因研究领域。（理由）

7.　许可证的发布

与程序有关的权利必须适用于该程序的任何使用者，并且程序的使用者也不需要为了使用该程序而获得其它许可证的许可。（理由）

8.　许可证不能针对于一个产品

与程序有关的权利不能由该程序是否作为某个软件产品的一部分来决定。如果程序从那个发布中被抽出来，并且按照程序的许可证的条款进行使用和发布，那么得到该程序的当事人或组织将获得与得到原程序的使用者相同的权利。（理由）

9.　许可证不能影响其它软件

许可证不得向与采用它的软件一同发布的其它软件提出任何限制。例如，许可证不能坚持要求在同一媒体上发布的其它程序都是开放源代码软件。（理由）

Conformance

（本节不是开放源代码定义的一个部分）

我们认为本开放源代码定义涵盖了由绝大多数软件团体使用的术语"开放源代码"的最初含义和当前含义。然而，该术语被广泛地应用，并且它的含义变得不精确了。OSI
Certified标志是某个软件发布许可证是否服从开放源代码定义的OST认证方式。一般的术语"开放源代码"并不提供这种担保，但我们仍然鼓励使用"开放源代码"这一术语以表明它符合"开放源代码定义"。关于OSI
Certified标志的信息，以及已经通过了OSI Certified、符合"开放源代码定义"的许可证，请参见OSI
Certified标志与纲要。

开放源代码定义之原理

给出开放源代码的定义的目的是：把我们所确信的、由软件开发团体所公认的"开放源代码"的含义作为一组具体的准则写下来---该准则确保按照开放源代码许可证发布的软件可以得到与其它软件同样认真的评审、使软件可以不断地得到改良和遴选，从而提供非开放软件所难以提供的可靠性与能力。

为了使此项工作持续发展，我们必须抵制人们为了短期利益而中止为软件开发做出贡献。这意味着，许可证的条款必须防止人们藏匿（lock
up）源代码从而导致只有很少的人才能够阅读和修改它。

当软件的开发者按照由OSI认证的许可证发布他们的软件时，他们可以在软件中使用"OSI Certified"标志。这种认证标志告知用户，该软件所采用的许可证符合开放源代码定义。关于我们的认证标志的更多信息及其纲要，请参见OSI
Certified标志与纲要。

1.　自由地再发布

通过强制要求许可证允许自由地再发布，我们抵制了任何为了获得少量短期销售金额而放弃长期效益的诱惑。如果我们不这样做，就会有很多压力迫使合作者放弃承诺。

2.　源代码

由于软件只有通过修改才能够得到改进，因此我们要求获得易于理解的源代码。因为我们的目的是使软件易于改进，我们也就希望软件易于修改。

3.　派生作品

仅仅提供可以阅读的源代码，对于实施认真的评审、对于软件的快速改良与遴选来说都是不够的。为了加快改进软件的速度，人们需要能够进行试验和对修改进行重新发布。

4.　作者的源代码的完整性

鼓励对软件进行各种改进是一件好事，但使用者有权知道谁应该对他们所使用的软件负责。软件的作者和维护者都有权知道需要他们提供支持的软件是什么，以维护他们的名誉。
因此，开放软件许可证必须保证可以容易地得到源代码，但可以要求源代码以原始源代码和一组补丁文件的形式发布。按照这种方式，就可以获得"非正式的"的修改而同时又可以轻易地把它们从原始源代码中区分出来。

5. 不得歧视任何个人或团体

为了从该进程中获得最大的收益，应该给尽可能多的个人和团体以同等资格对开放源代码软件做出贡献。为此我们禁止开放源代码许可证把任何人排除在这个进程之外。

6. 不得歧视任何应用领域（fields of endeavor）

该条款的主要目的是禁止许可证中含有使开放源代码软件无法在商业上使用的规定。我们需要商业用户参与我们的工作，而不让他们感到被排除在外。

7. 许可证的发布

本条款的目的是禁止通过非直接的手段，例如需要签署非公开的协定，使软件无法公开。

8. 许可证不能针对于一个产品

本条款用于预防其它许可证陷阱。

9. 许可证不能影响其它软件

开放源代码软件的发布者有权为他们自己的软件做出选择。
是的，GPL满足这条要求。按GPL发布的库只影响（contaminate）在运行时
与之连接的软件，于仅仅和它一同发布的软件无关。

OSI Certified标志与纲要

开放源代码定义给出了开放源代码软件的基本性质。不幸的是，术语"开放源代码"遭到了滥用，并且由于它的描述性，它不能作为一个商标（这是我们的首选）被保护。由于社团需要一种可靠的方式以确定一份软件是否真正是开放源代码软件。OSI为此目的而注册了一个认证标志，OSI Certified。如果你在一份软件上看到了该标志，那么该软件就是必定是按照遵从开放源代码定义的许可证发布的，否则，该发行人就是在滥用该标志而且违反了法律。
OSI Certified标志被用于软件而不是许可证：人们需要知道的是组成软件的包和它们的许可证是开放的源代码。此外，单独的许可证可能不是"货物"，它由美国专利和商标局注册认证标志。但是软件的作者显然必须能够在不询问我们的情况下（自我认证）、在适当的时刻确认他们的软件是按照OSI Certified的开放源代码方式发布的。所以认证按照两步进行：
* OSI提供一个满足开放软件源代码定义的开放源代码许可证表。这些许可证经过公众的详细审查并且被我们所批准。如果你有一个希望加入该列表的许可证，请与

license-approval@opensource.org联系，它启动了获得许可证批准的程序。
* 如果你希望在你的软件上设置OSI Certified标志，你可以从列表中选择一个获得批准的许可证并根据它发布软件，并且适当地为软件做标志。
上面仅仅是一个概括。下面给出申请批准许可证和把OSI Certified标志应用在你的软件上的详细说明。

获得许可证的批准

1. 通过电子邮件把许可证发送给license-approval@opensource.org。在电子邮件中说明你是否愿意以你的签名或者匿名地把许可证发送到许可证讨论列表中。（我们愿意考虑那些根本不希望被发送的许可证，但由于社团的评审是批准的一个重要组成部分，我们将不得不把该许可证私下地发送给评审者：因此，对没有被发送到许可证讨论列表中的许可证的批准，要花费更长的时间，并且通常要更多地与你交流。）

2. 如果我们发现你的许可证不符合开放源代码的定义，我们将与你一同解决这个问题。
3. 同时，我们将关注许可证论坛列表，并且与你一同工作以解决大家提出的任何未包含的问题。
4. 作为该过程的一部分，我们还将就许可证问题向外界寻求法律上的建议。
5. 一旦许可证符合了开放源代码定义，并且在许可证论坛上经过了充分的讨论或者其它的评审者

没有提出我们认为重要的问题，我们将通知你，许可证已经被批准了，同时它被复制到我们的网站上，并且被加入以下的许可证列表。

使用该标志

你可以在这两种软件的发行上使用OSI Certified标志。该软件的发布包括并且达到了被批准的许可证列表中某个许可证的要求，或者该软件的源代码已经被明确地放到公众可以得到的位置。

为了表明你的软件是按照OSI Certified发布的，你必须不加任何修改地把下面两者之一，按照下述方式加入你的软件。完整的声明是：

This software is OSI Certified Open Source Software.

OSI Certified is a certification mark of the Open Source Initiative.

较简短的声明是：

OSI Certified Open Source Software

你的软件的每种发布形式都有自身的要求：

* 如果软件以电子的形式发布（而不是实物（tangible）的形式），你必须把完整的声明放在README文件或者类似的、人类用户将首先阅读的文件中。

* 如果软件以实物形式发布，你必须完成以下所有可行的任务：

. 如果软件的发布包括印刷品，你必须把完整的声明加入该印刷品。

. 如果软件发布在可擦写的媒体（例如软磁盘）、CD-ROM、磁带等媒体上，它们在物理上可以添加能够被人的裸眼所阅读的简短声明，而不影响它们的功能，你必须把完整的声明或简短的声明标注在该媒体上。

. 如果软件的发布是一个的包，从而使其中的声明不能被阅读。你必须在包的外面放置完整的声明。

如果不能把以上任何条款用于发布，请与我们联系，我们将把你选择的位置加入本列表。

获得批准的许可证

以下许可证满足开放源代码的定义，并且已经被批准用于OSI Certified的开放源代码软件。没有给出批准日期的许可证是在1999年1月1日以前批准的。

* The GNU General Public License（GPL）;

* The GNU Library（Lesser） General Public License（LGPL）;

* The BSD License;

* The X Consortium License;

* The Artistic License;

* The Mozilla Public License（MPL）;

* The QPL;

其它符合定义的许可证包括：libpng许可证、zlib许可证、IJG JPEG许可证和OPL（Open LDAP Public
License）许可证。

如何通过搜索找到你想要的ISO

我们一起在风雨中前行

如何通过搜索找到你想要的ISO

最近很多兄弟在论坛上问到关于在什么地方下载某一版本的ISO，古话说得好授之与雨不入授之以渔。正好乘着在加闲着写篇文字教大家如何在网络上找到自己想要的东西。大家一看到我上面的那句话，我想大半就知道我想要说什么了。今天我不向各位介绍163搜索
也不向大家介绍新浪搜索。今天我要向大家介绍的是北大天网搜索引擎。北大天网最软件的搜索非常有效，特别是他能够搜索ftp服务器，这样也就我们寻找iso一个非常行之有效的方法。具体的介绍请大家到了北大天网在详细了解。现在我简要介绍下使用方法：

一：登陆到北大天网网站，网址为 http://e.pku.edu.cn如下图

二：在搜索栏目中输入你想要找的iso的描述，例如我要找SuSe8.0我可以输入suse 要找Red Hat7.3就可以输入redhat
等等如此。如下图

三：此时也许你就可以马上找到你想要的，也有可能找非常多的结果 往往达到上万条。如下图

见到这么多的结果你不要气馁。北大天网拥有一个功能就是在你搜索到的结果进一不搜索。如上图

四：我想此时你应该使会看到你想要找的结果，例如我刚才查找suse8.0的结果。如下图

如果第四步的结果是没有找到任何iso，这个时候你不要放弃。你可以试着改变你的查找顺序。再以suse8.0为例，我可以先找suse再进一步找8.0；你也可以先找8.0再从搜索的结果中搜索suse。在这里搜索的方式是可以灵活使用的。希望这段文字能够在你学习linux的过程中给以帮助。

linux是我们的linux，让我们一同迈着坚定的脚步走向windows的灭亡！感谢那些为linux发展默默无闻的程序员们以及一切关心linux成长的人们.

什么是Linux

我们一起在风雨中前行

什么是Linux

简单的说,Linux是Unix克隆(Unix clone)或Unix风格(Unix alike)的操作系统(OS),在原代码级上兼容绝大部分Unix标准(指的是IEEE
POSIX,System V,BSD),是一个支持多用户, 多进程,多线程,实时性较好的功能 强大而稳定的操作系统.它可以运行在x86
PC,Sun Sparc,Digital Alpha,680x0,PowerPC, MIPS等平台上,可 以说Linux是目前运行硬件平台最多的操作系统.
Linux最大的特点在于它是GNU(Gnu's Not Unix----有点分形与混沌的意味----无限自包含,简单的说GNU是一种自由软件体系)的一员,遵循公共版权许可证(GPL),秉承
"自由的思想,开放的源码"的原则,成千上万的专家/爱好者通过Internet在不断地完善并维护它,可以说Linux是计算机爱好者自己的操作系统.

追述Linux的历史直到1990年,Linus Torvalds还是芬兰赫尔辛基大学的一名学生,最初是用汇编语言写了一个在80386保护模式下处理多任务切换的程序,后来从Minix(Andy
Tanenbaum教授所写的很小的Unix操作系统,主要用于操作系统教学)得到灵感,进一步产生了自认为狂妄的想法----写一个比Minix更好的Minix,于是开始写了一些硬件的设备驱动程序,一个小的文件系统,......,这样0.0.1版本的Linux就出来了,但是它只具有操作系统内核的勉强的雏形,甚至不能运行,你必须在有Minix的机器上编译以后才能玩.这时候Linus已经完全着迷而不想停止,决定踢开Minix,于是在1991年10月5号发布Linux
0.0.2版本,在这个版本中已经可以运行bash(the GNU Bourne Again Shell----一种用户与操作系统内核通讯的软件)和gcc(GNU
C 编译器).从一开始,Linus就决定自由扩散Linux,包括原代码,他在comp.os.minix新闻讨论组里发布Linux 0.0.2时写到:

"Do you pine for nice days of Minix-1.1, when mem were men
and wrote their own device drivers? Are you without a nice project
and just dying to cut your teeth on a OS you can try to modify for
your needs? Are you finding it frustrsting when everything works on
Minix? No more all-nighters to get a nifty program working? Then this
post might be just for you. "As I mentioned a month ago, I'm
working on a free version of a Minix-lookalike for AT-386 computers.
It has finally reached the stage where it's even usable(though may
not be depending on what you want),and I am willing to put out the
sources for wider distribution. It is just version 0.0.2 ... but I've
successfully run bash,gcc,gnu-make,gnu-sed,compress,etc.under it."

随即Linux引起黑客们(hacker)的注意,通过计算机网络加入了Linux的内核开发,Linux倾向于成为一个黑客的系统----直到今天,在Linux社区里内核的开发被认为是真正的编程.由于一批高水平黑客的加入,使Linux发展迅猛,到1993年底94年初,Linux
1.0终于诞生了! Linux 1.0已经是一个功能完备的操作系统,而且内核写得紧凑高效,可以充分发挥硬件的性能,在4M内存的80386机器上也表现得非常好,至今人们还在津津乐道于此,不过自从2.1.xx系列的内核Linux开始走高端的路子----硬件的发展太快了,但是Linux不会失去它的本色.Linux具有良好的兼容性和可移植性,大约在1.3版本之后,开始向其他硬件平台上移植,包括号称最快的CPU---Digital
Alpha(至少目前主频是最高的).所以不要总把Linux与低档硬件平台联系到一块,Linux发展到今天,这是一个误区,它只是将硬件的性能充分发挥出来而已,Linux必将从低端应用横扫到高端应用!

在Linux的发展历程上还有一件重要的事:Linux加入GNU并遵循公共版权许可证(GPL).此举大大加强了GNU和Linux,几乎所有应用的GNU库/软件都移植到Linux,完善并提高了Linux的实用性,而GNU有了一个

根基,我现在也搞不清楚到底是GNU
Linux呢还是基于Linux的GNU. :-) 更重要的是遵循公共版权许可证,在继承自由软件的精神的前提下,不再排斥对自由软件的商业行为(如把自由软件打包以光盘形式出售),不排斥商家对自由软件进一步开发,不排斥在Linux上开发商业软件.从此Linux又开始了一次飞跃,出现了很多的Linux发行版,如Slackware,Redhat,Suse,TurboLinux,OpenLinux等十多种,而且还在增加,注意你不能说"Redhat Linux""Suse Linux""Slackware Linux"等等,Linux主要指操作系统内核,对所有发行版内核原代码都是一样的(但集成的内核版本可能因发行时间不同而有所不同).还有一些公司在Linux上开发商业软件或把其他Unix平台的软件移植到Linux上来,如今很多IT业界的大腕如
IBM,Intel,Oracle,Infomix,Sysbase,Corel,Netscape,CA,Novell等都宣布支持Linux!
商家的加盟弥补了纯自由软件的不足和发展障碍,Linux迅速普及到广大计算机爱好者,并且进入商业应用,正是打破某些公司垄断文化圈的希望所在!!

Linux是爱好者们通过Internet协同开发出来的,当然它的网络功能十分强大,比如你可以通过ftp,nfs等来安装Linux,用它来做网关等等.随着Linux的发展衍生出来的应用恐怕出乎Linus本人最初的预料,如有人用它来做路由器,有人来做嵌入式系统,有人来做实时性系统......常有新手问Linux能做什么,其实它不象那些中看不中用的操作系统,不在于你用它能干什么,而在于你想干什么!

Linux的兴起还给人们很多启迪与思考,如集市式软件开发的讨论,又如自由软件的精神......

2005-1-20 11:27:26 上一页 首页 下一页

2005-1-20 上一页 首页 下一页

问题解决之道

我们一起在风雨中前行

问题解决之道

我想无论你到什么论坛的那个版块的最顶端你都会看到下面这样一篇帖子，这个帖子就如何合理的使用论坛给出了几点的意见，希望这个对大家以后学习linux能够有一点的帮助。（本贴来自linux论坛）

一、发帖子时，先在论坛里搜索一下有没有相关的帖子。这样我们就提高了解决问题的效率，以免重复

发帖。我常在国外网站发帖， 因为原来不懂这个规矩，我收到了好几个警告，有的是网站管理人员的，大多的是网站的会员。但网站的管理员还是把的帖子重新编辑，放在相应的位置上，并发一封信告诉帖子的位置，以及发帖子的方法和技巧等。在这方面，我们还要多学习一下。

二、我们发帖子时，要写好主题标题，简洁明了，切重要点。最好不用［紧急、请帮助、求助、特急之类的词语］，因为我们的帖子是给答复我们的弟兄看的，也是给有此问题的朋友看的。如果你用这些词语，可能有不会此问题的朋友也要看,这样就浪费了这些弟兄的时间,另外如果他们也有此问题，他们也搜索不到你的主题。有此经验的弟兄也能立即答复你的问题。我们应该用与我们问题相关的词语。如果你的SAMBA有问题，那就在主题上写［SAMBA问题］
。

三、尽可能不要用与主题无关的图片。如果确实要上传与主题相关的图片，请尽量提供联接，这样可以节约更多的空间。

四、详细说明你的情况，如果你要问硬件问题，你就得把相关硬件的参数、品牌、芯片组等，以及你设置相关的参数出现的错误信息等。同时也得把系统说上，因为每个系统的配制方法都有不同的地方，这个极重要，否则弟兄们真的不知道怎么回答。

五、如果你问的是其它的问题，你就得说出这类错误的详细信息。

六、如果你原来已经发过你所要问的问题，如果弟兄们给你的答复，你仍没有解决了，那你最好还接着原来的问题再发帖子。如果别人也有相似的问题，你也可以跟帖子，因为类似的问题一集中，我们就容易解决。

七、发帖时要讲究说话方式，因为在咱们的论坛里，弟兄们能不能帮助你，不是弟兄们的义务，是弟兄们自愿的。所以我们提问时尽可能的客气一点。用语讲究客气，能为我们营造了一个良好的互助氛围。这对我们每个弟兄都一样，因为我们不是什么都会的。

八、要有爱心，有奉献精神。因为我们面对的是大多的从WINDOWS转过来的弟兄，他们习惯于点鼠标，可能暂时不太习惯LINUX的操作方式。这就要我们给他以信心，让他有勇气敢于挑战LINUX给他带来的不便。我们要学会怎样倾听发帖提问的弟兄们的心理感受。我们都知道，我们学WINDOWS时也是一样的，点几下鼠标，有时也问好几个人，读好多书，其实真正有问题要问的还是LINUX，而不是WINDOWS。如果我们认真的彼此关注，我们就会感受到“互帮助给予我们的快乐和感动”。我们每个人都需要帮助，我们每个人都要尽可能的为别人做点什么，没有付出就没有收获。

九、发帖时想一下你的主题应该发到哪个版块。这样才能让兄弟们在最短的时间内找到所需要的信息，事半功倍嘛。

十、归纳总结，如果问题解决了，我们最好写个总结，把你的心得和成果发布出来，这样也便于我们帮助有此问题的兄弟。

2005-1-20 11:27:26

目录

GRUB三步通

我们一起在风雨中前行

GRUB三步通
##################
#　　GRUB的优点　　#
##################
　　GRUB 是引导装入器(boot loader) -- 它负责装入内核并引导 Linux 系统。GRUB
还可以引导其它操作系统，如 FreeBSD、NetBSD、OpenBSD、GNU HURD 和 DOS，以及 Windows 95、98、NT
和 2000。尽管引导操作系统看上去是件平凡且琐碎的任务，但它实际上很重要。如果引导装入器不能很好地完成工作或者不具有弹性，那么就可能锁住系统，而无法引导计算机。另外，好的引导装入器可以给您灵活性，让您可以在计算机上安装多个操作系统，而不必处理不必要的麻烦。

GRUB 是一个很棒的boot loader。它有许多功能，可以使引导过程变得非常可靠。例如，它可以直接从 FAT、minix、FFS、ext2
或 ReiserFS 分区读取 Linux 内核。这就意味着无论怎样它总能找到内核。另外，GRUB 有一个特殊的交互式控制台方式，可以让您手工装入内核并选择引导分区。这个功能是无价的：假设
GRUB 菜单配置不正确，但仍可以引导系统。哦，对了 -- GRUB 还有一个彩色引导菜单。更令人惊讶的是，这是一个自由软件!
##################
#　　 GRUB菜单　　 #
##################
　　先来看一个例子，这是位于/boot/grub/目录下的menu.lst文件。
此文件将在开机是产生一个菜单，包含有Debian linux,Windows2000,RedHat linux和 Mandrake linux,

共四个选择项。我一共分了8个区，一个fat16（0x6），一个ntfs（0x7），三个ext2fs分区（0x83），一个swap分区（0x82）。ntfs用来装win2000，三个ext2fs装了三个linux，c盘fat16分区没有装任何东西。

```
＃例子由此开始
＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃
#       一个GRUB configure 的例子        #
＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃
timeout 10
default 2

# --> Debian linux <--
title Debian linux
root (hd0,2)
kernel /boot/vmlinuz-2.2.18 root=/dev/hda3 ro
initrd /boot/initrd-2.2.18.gz

# --> Debian END <--
# --> Windows 菜单选项 <--
title Windows2000
root (hd0,0)
chainloader +1
# --> Winddows 结束 <--
# --> RedHat linux 菜单选项 <--
title RedHat linux
root (hd0,8)
chainloader +1 # 在硬盘主引导分区装了lilo，所以也用了chainloader。
# --> RedHat linux 结束 <--
# --> Mandrake linux 菜单选项 <--
title Mandrake linux
root (hd0,5)
kernel /boot/vmlinuz-2.4.3-20mdk root=/dev/hda6 ro
initrd /boot/initrd-2.4.3-20mdk.img
# --> Mandrake linux 结束 <--

＃例子到此结束
```

(以符号井"＃"开头的行表示被注释掉，没有任何意义。)

timeout表示默认等待的时间，这儿是10秒钟。超过10秒，用户还没有作出选择的话，系统将自动选择默认的操作系统。

默认的操作系统就是由default控制的。default后加一个数字n，表明是第n＋1个。需要注意的是，GRUB中，计数是从0开始的，第一个硬盘是hd0，第一个软驱是fd0，等等。所以，default
2 表示默认的操作系统在这儿是 Redhat linux。

接下来，正如你所想象的，title表示的是“Debian linux”菜单项。root (hd0,2)表示第一个硬盘,第三个分区。这儿的root
于linux的root分区及其不同，此root非彼root也！

在 Linux 中，当谈到 ″root″ 文件系统时，通常是指主 Linux 分区。但是，GRUB 有它自己的
root

分区定义。GRUB 的 root 分区是保存 Linux 内核的分区。这可能是您的正式 root 文件系统，也可能不是。我们讨论的是 GRUB，需要指定
GRUB 的 root 分区。进入 root 分区时，GRUB 将把这个分区安装成只读型，这样就可以从该分区中装入 Linux 内核。GRUB
的一个很“酷”的功能是它可以读取本机的 FAT、FFS、minix、ext2 和 ReiserFS 分区。

到目前为止，您可能会感到一点疑惑，因为 GRUB 所使用的硬盘 / 分区命名约定与 Linux 使用的命名约定不同。在 Linux 中，第一个硬盘的第五个分区称作
"hda5"。而 GRUB 把这个分区称作 "(hd0,4)"。GRUB 对硬盘和分区的编号都是从
0 开始计算。另外，硬盘和分区都用逗号分隔，整个表达式用括号括起。现在，可以发现如果要引导 Linux 硬盘 hda5，应输入 "root
(hd0,4)"。

知道了内核在哪儿，还要具体指出哪个文件是内核文件，这就是kernel的工作。
kernel /boot/vmlinuz-2.2.18 root=/dev/hda3 ro说明/boot/vmlinuz-2.2.18 就是要载入的内核。后面的都是传递给内核的参数。root=/dev/hda3就是linux的硬盘分区表示法，ro是以readonly的意思。

initrd用来初始的linux image，并设置相应的参数。

是不是感觉很简单啊！再来看一看windows的定义段吧。

这里，我添加了一项来引导 Windows2000。要完成此操作，GRUB 使用了“链式装入器”(chainloader)。链式装入器从分区
(hd0,0) 的引导记录中装入 win2000 自己的引导装入器，然后引导它。这就是这种技术叫做链式装入的原因 -- 它创建了一个从引导装入器到另一个的链。这种链式装入技术可以用于引导任何版本的
DOS 或 Windows。

我的RedHat linux在硬盘主引导分区装了lilo，所以也用了chainloader。

GRUB的配置文件要简单就这么简单，如果你要更个性化一点，试一试把“color light-gray/blue ”加在default语句的下面，下一次启动GRUB时，看看有什么变化，再试一试“color light-blue/red",惊喜吗？ 有趣吧!

######################
GRUB的交互性
######################

GRUB 最好的优点之一就是其强健的设计 -- 在不断使用它时请别忘了这点。如果更新内核或更改它在磁盘上的位置，不必重新安装
GRUB。事实上，如有必要，只要更新 menu.lst 文件即可，一切将保持正常。

只有少数情况下，才需要将 GRUB 引导装入器重新安装到引导记录。首先，如果更改 GRUB root 分区的分区类型（例如，从 ext2
改成 ReiserFS），则需要重新安装。或者，如果更新 /boot/grub 中的 stage1 和 stage2 文件，由于它们来自更新版本的
GRUB，很有可能要重新安装引导装入器。其它情况下，可以不必理睬！

GRUB的最大的特点就是交互性特别强.在开机时，按一下“c”，将进入GRUB 控制台.显示如下：

```
GRUB version 0.5.96.1 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the
first word, TAB
lists possible command completions. Anywhere else TAB lists the possible

completions of a device/filename. ]
grub>
```

欢迎使用 GRUB 控制台。现在，再研究命令：

我将通过GRUB 控制台绕过lilo来启动RedHat linux，
grub> root (h
现在，按一次 Tab 键。如果系统中有多个硬盘，GRUB 将显示可能完成的列表，从 "hd0"
开始。如果只有一个硬盘，GRUB 将插入 "hd0,"。如果有多个硬盘，继续进行，在 ("hd2")
中输入名称并在名称后紧跟着输入逗号，但不要按 Enter 键。部分完成的 root 命令看起来如下：
grub> root (hd0,
现在，继续操作，再按一次 Tab 键。GRUB 将显示特定硬盘上所有分区的列表，以及它们的文件系统类型。在我的系统中，按
Tab 键时得到以下列表：
grub> root (hd0, (tab，按tab一下键)
Possible partitions are:
Partition num: 0, Filesystem type is fat, partition type 0x6
Partition num: 2, Filesystem type is ext2fs, partition type 0x83
Partition num: 4, Filesystem type unknown, partition type 0x7
Partition num: 5, Filesystem type is ext2fs, partition type 0x83
Partition num: 6, Filesystem type is fat, partition type 0xb
Partition num: 7, Filesystem type is fat, partition type 0xb
Partition num: 8, Filesystem type is ext2fs, partition type 0x83
Partition num: 9, Filesystem type unknown, partition type 0x82

如您所见，GRUB 的交互式硬盘和分区名称实现功能非常有条理。这些，只需要好好理解 GRUB 新奇的硬盘和分区命名语法，然后就可以继续操作了

grub> root (hd0,8)
现在已安装了 root 文件系统，到装入内核的时候了
grub> kernel /boot/vmlinuz-2.4.2 root=/dev/hda5 ro

[Linux-bzImage, setup=0x1200, size=0xe1a30]
您已经安装了 root 文件系统并装入了内核。现在，可以引导了。只要输入 "boot"，Linux 引导过程就将开始。是不是很cool啊，GRUB的menu.lst更像一个linux下的脚本程序。

#####################
GRUB启动盘
#####################
要制作引导盘，需执行一些简单的步骤。首先，在新的软盘上创建 ext2 文件系统。然后，将其安装，并将一些
GRUB 文件复制到该文件系统，最后运行 "grub" 程序，它将负责设置软盘的引导扇区。准备好了吗？
将一张空盘插入 1.44MB 软驱，输入：
mke2fs /dev/fd0
创建了 ext2 文件系统后，需要安装该文件系统：
mount /dev/fd0 /mnt/floppy
现在，需要创建一些目录，并将一些关键文件（原先安装 GRUB 时已安装了这些文件）复制到软盘：
mkdir /mnt/floppy/boot
mkdir /mnt/floppy/boot/grub
cp /boot/grub/stage1 /mnt/floppy/boot/grub
cp /boot/grub/stage2 /mnt/floppy/boot/grub
再有一个步骤，就能得到可用的引导盘。

在linux bash中，从 root 用户运行“grub”，该程序非常有趣并值得注意，因为它实际上是 GRUB
引导装入器的半功能性版本。尽管 Linux 已经启动并正在运行，您仍可以运行 GRUB 并执行某些任务，而且其界面与使用 GRUB 引导盘或将
GRUB 安装到硬盘 MBR 时看到的界面(即GRUB控制台)完全相同。

在 grub> 提示符处，输入：

grub> root (fd0)

grub> setup (fd0)

grub> quit

现在，引导盘完成了。

如果要把GRUB装到硬盘上，也很容易。这个过程几乎与引导盘安装过程一样。首先，需要决定哪个硬盘分区将成为 root GRUB 分区。在这个分区上，创建
/boot/grub 目录，并将 stage1 和 stage2 文件复制到该目录中，可以通过重新引导系统并使用引导盘，或者使用驻留版本的
GRUB 来执行后一步操作。在这两种情况下，启动 GRUB，并用 root 命令指定 root 分区。例如，如果将 stage1 和 stage2
文件复制到 hda5 的 /boot/grub 目录中，应输入 "root (hd0,4)"。接着，决定在哪里安装 GRUB
-- 在硬盘的 MBR，或者如果与 GRUB 一起使用另一个“主”引导装入器，则安装在特定分区的引导记录中。如果安装到 MBR，则可以指定整个磁盘而不必指定分区，如下（对于
hda）：

grub> setup (hd0)

如果要将 GRUB 安装到 /dev/hda5 的引导记录中，应输入：

grub> setup (hd0,4)

现在，已安装 GRUB。引导系统时，应该立即以 GRUB 的控制台方式结束（如果安装到 MBR）。现在，应创建引导菜单，这样就不必在每次引导系统时都输入那些命令。

小结：在这里只是介绍了 GRUB 的一部分。例如，可以使用 GRUB 来执行网络引导，引导 BSD 文件系统，或更多操作。另外，GRUB
有许多配置和安全性命令也很有用。如需所有 GRUB 功能的完整描述，请阅读 GRUB 出色的 GNU 文档。只要在 bash 提示中输入 "info
grub" 就可以阅读该文档。

有用的一些信息：xosl是一个支持鼠标的图形界面boot loader，可以于system command和boot manager相比网址是http://www.xosl.org

GRUB的下载：ftp://alpha.gnu.org/gnu/grub/

Linux init详解

我们一起在风雨中前行

一、什么是INIT:

init是Linux系统操作中不可缺少的程序之一。

所谓的init进程，它是一个由内核启动的用户级进程。

内核自行启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序init的方式，完成引导进程。所以,init始终是第一个进程（其进程编号始终为1）。

内核会在过去曾使用过init的几个地方查找它，它的正确位置（对Linux系统来说）是/sbin/init。如果内核找不到init，它就会试着运行/bin/sh，如果运行失败，系统的启动也会失败。

二、运行级别

那么，到底什么是运行级呢?

简单的说，运行级就是操作系统当前正在运行的功能级别。这个级别从1到6 ，具有不同的功能。

不同的运行级定义如下：(可以参考Red Hat Linux 里面的/etc/inittab)

0 - 停机（千万不能把initdefault 设置为0 ）

1 - 单用户模式

2 - 多用户，没有 NFS

3 - 完全多用户模式(标准的运行级)

4 - 没有用到

5 - X11 （xwindow)

6 - 重新启动 （千万不要把initdefault 设置为6 ）

这些级别在/etc/inittab 文件里指定。这个文件是init 程序寻找的主要文件，最先运行的服务是放在/etc/rc.d
目录下的文件。在大多数的Linux 发行版本中，启动脚本都是位于 /etc/rc.d/init.d中的。这些脚本被用ln 命令连接到 /etc/rc.d/rcn.d
目录。(这里的n 就是运行级0-6)

三、运行级别的配置

运行级别的配置是在/etc/inittab行内进行的，如下所示：

12 : 2 : wait : / etc / init.d / rc 2

第一个字段是一个任意指定的标签；

第二个字段表示这一行适用于运行那个级别（这里是2）；

第三个字 段表示进入运行级别时，init应该运行第四个字段内的命令一次，而且init应该等待该命令结束。/etc/init.d/rc命令运行启动和终止输入以便进入运行级别2时所需的任何命令。

第四个字段中的命令执行设置运行级别时的一切“杂活”。它启动已经没有运行的服务，终止不应该再在新运行级别内运行的服务。根据Linux版本的不同，采用的具体命令也不同，而且运行级别的配置

也是有差别的。

init启动时，它会在/etc/inittab内查找一个代码行，这一行指定了默认的运行级别：

id : 2 : initdefault :

你可以要求init在启动时，进入非默认运行级别，这是通过为内核指定一个“single”或
“emergency”
命令行参数来实现的。比如说，内核命令行参数的指定可通过LILO来执行。这样一来，你就可以选择单用户模式了（即运行级别1）。

系统正在运行时，telinit命令可更改运行级别。运行级别发生变化时， init 就会从/etc/inittab运行相应的命令。

四、/etc/inittab中的特殊配置

/etc/inittab中，有几个特殊的特性，允许init重新激活特殊事件。这些特殊特性都是用第三个字段中的特殊关键字标记出来的。比如：

1. powerwait

允许init在电源被切断时，关闭系统。其前提是具有U P S和监视U P S并通知init电源已被切断的软件。

2. ctrlaltdel

允许init在用户于控制台键盘上按下C t r l + A l t + D e l组合键时，重新启动系统。注意，如果该系统放在一个公共场所，系统管理员可将C
t r l + A l t + D e l组合键配置为别的行为，比如忽略等。

3. sysinit

系统启动时准备运行的命令。比如说，这个命令将清除/tmp。

上面列出的特殊关键字尚不完整。其他的关键字及其使用详情，可参考你的inittab手册页。

五、在单用户模式下引导

一个重要的运行级别就是单用户模式（运行级别1），该模式中，只有一个系统管理员使用特定的机器，而且尽可能少地运行系统服务，其中包含登录。单用户模式对少数管理任务（比如在/usr分区上运行fsck）而言，是很有必要的，因为这需要卸载分区，但这是不可能的，除非所有的服务系统已被杀死。

一个正在运行的系统可以进入单用户模式，具体做法是利用init，请求运行级别1。内核启动时，在内核命令行指定single或emergency关键字，就可进入运行级别1了。内核同时也为init指定命令行，init从关键字得知自己不应该采用默认的运行级别（内核命令行的输入方式和你启动系统的方式有关）。

有时，以单用户模式进行启动是必要的，这样一来，用户在装入分区之前，或至少在装入分散的/usr分区之前，能手工运行fsck（在分散的文件系统上，任何活动都可能使其更为分散，所以应该尽可能地运行fsck）。

如果自动化的fsck在启动时失败了，启动脚本init的运行将自动进入单用户模式。这样做是为了防止系统使用不连贯的文件系统，这个文件系统是f
s c k不能自动修复的。文件系统不连贯的现象极为少见，而且通常会导致硬盘的不连贯或实验性的内核释放，但最好能做到防患于未然。

由于安全上的考虑，在单用户模式下，启动外壳脚本之前，配置得当的系统会要求用户提供root密码。否则，它会简单地为L
I L O输入合适的一行代码，以r o o t的身份登录（当然，如果/etc/passwd已经由于文件系统的问题而不连贯了，就不适合这里的原则了，为对付这种情况，你最好随时准备一张启动盘）。

不同的运行级有不同的用处，也应该根据自己的不同情形来设置。

例如，如果丢失了root口令，那么可以让机器启动进入单用户状态。在启动后的 lilo 提示符下输入：

init=/bin/sh rw 使机器进入运行级1 ，并把 root 文件系统挂为读写。他会跳过所有系统认证，让你可以使用passwd
程序来改变root口令，然后启动到一个新的运行级。

linux常用命令索引

我们一起在风雨中前行

linux常用命令索引

在Linux中有近千条的命令，由于篇幅的关系，不可能将所有的命令全部列出。本附录只是对一些常用的命令进行介绍。读者在使用没有列出的命令时，如果有任何疑问，可以使用“command--help”获得该命令的在线帮助，或着查看相关的manpages获得更详细地使用说明。

alias

用一个简单的名字来代替一条复杂的命令（起别名），如果忽略了命令名，则打印所有已定义的命令的别名。

语法：

alias[参数]aliasname=’命令’

-t为Linux命令创建一个可跟踪的别名。

-x输出别名，使之在shell脚本中使用。

ash

由KennethAlmquist在1989年编写，它的很多特性接近于SYSTEMV的shell。

语法：
ash[-eflijnsxz][+eflijnsxz][-c命令][参数]
-c若使用-c参数，则ash在执行完这条命令后退出。

-s若用-s参数，则ash从标准输出中读入命令，如果不跟-c-s参数则ash以所跟的第一个参数为文件名，从此文件中读入命令，直接输出到exit。如果参数0的第一个字符是“-”则ash确定为loginshell，ash将从/etc/profile或用户目录中的.profile读入相应的设置和环境变量。
-e执行命令后返回值为非零值时退出ash。
-f关闭ash自动产生文件名的功能。
-j打开伯克利Unix风格的工作控制。
-n读入命令但不执行。
banner
在输出行中打印高质量的标题，如果输出要打印的文字，则其等待从标准输入中输入一行文字。
语法：
/usr/games/banner[-wn]msg
-w宽度。输出宽度从132到n，n默认值为80列
bash
是目前广泛使用的一种shell，同sh兼容，并且包含了ksh和csh中一些有用的特性。
语法：
bash[参数][文件名]
-c字符串从字符串中读入命令，如果字符串后还有变量则被设定为从$0开始的位置参数。
-i交互式的执行命令。例如：rm－i。
-s从标准输入中读命令，直到输入exit。
-表示参数执行完毕，并且屏蔽此后所跟的参数，后面的所有变量都被看作是文件名。
-norc如果bash是交互的，则不执行个人初始化文件~/.bashrc，如果bash作为shell来运行，这个参数缺省是关闭的。

-noprofile不执行系统范围内的启动文件，也不执行个人启动文件~/.bash_profile、~/bash_login或~/.profile，缺省情况下，bash作为登录shell时以这些文件作为启动文件。
-rcfile文件名如果bash是交互的，则以此文件作为bash的启动文件。
-version在bash开始时显示此bash的版本号。
-quiet不显示版本号或其它信息，这是默认值。
-login激活bash伪装为登录shell。
-posix改变bash的行为，使其符合Posix1003.2规定的标准。
cal
显示一个日历。对于查询某年某月某日的用户来说是非常有用的。
语法：
cal[-jy][月年]
-jx显示julian日期（日子是从1月1日开始累加的）
-y显示今年的日历。

年份必需写全，“cal99”是不正确的，应该写为“cal1999”。月份必须是1-12之间的数字或英文全称。不加任何参数则输出本月日历。
cat

将文件连接到一起，用“>”操作将几个文件组合成一个新的文件，或者是用“>>”把文件追加到已存文件的后面。
语法：
cat[参数]filelist
-e打印一个$来表示每行的结尾，这个选项只有与-v选项一起使用才有效。
-v显示控制字符或其它不可打印的字符。
-s隐去不可读文件的信息。
-t把每个tab打印为^l，并把Formfeed用^L填充
-u无缓冲的打印输出。
filelist用于结合文件的可选列表。
cd
改变当前工作目录。
语法：
cd目录名
如果后面不加目录名，则返回用户的根目录，此外用户必须拥有进入该目录的权限。
chgrp

该变文件的组所有权。一些chgrp命令的选项有长短两种选项格式。我们这里只列出这些选项的短表示方式。详细资料请参看manpages。
语法：
chgrp[参数]组文件
-c只有在文件的所有权确实改变时才进行详细地说明。
-f不打印所有权不能该变的文件的错误信息。
-v详细说明所有权的变化。
-R递归改变目录及其内容的所有权
--help在标准输出上打印使用信息并退出。
--version在标准输出行上打印版本信息并退出。
group要改变的组。可以是GID，也可以是有效组名。
files要改变组所有权的文件列表，以空格分开。
若不是文件的所有者或root，则不能修改文件的组所有权。
chmod

改变文件的模式。文件模式控制了与该文件相关的访问权限，Linux有三个安全级别：所有者级、组访问级和其它用户级。在这三个级中，有三种权限：读、写和执行。对于标准文件，读权限意味着可以浏览文件的内容，写权限可以修改文件，执行权限可以运行该文件。目录形式有所不同。读权限可以浏览该目录下的内容，写权限可以在目录下创建文件或删除文件。执行权限意味着可以从一个目录转变到另一个目录。
语法：
chmod[参数]保护模式文件或目录
-c只有在文件的权限确实改变时才进行详细地说明。
-f不输出权限不能改变的错误信息。
-v详细说明权限的变化。
-R改变本目录及其所有子目录的文件的权限。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。

保护模式格式为[ugoa…][[+-=][rwxXstugo…]…][,…]
cp
拷贝文件。
语法：
cp[参数]源文件目标文件或目录
-a在备份中尽可能的保留源文件的结构和属性。
-b做将要覆盖或删除文件的备份。
-d将符号链接作为符号链接拷贝，而不拷贝它们所链接的文件。并在备份中保留源文件间固有的链接关系。
-f删除已存在的目标文件。
-i提示是否覆盖已存在的目标文件。
-l形成固有链接以替代非目录的拷贝。
-P通过加入目标目录分支和指定的源文件名形成每个目标文件名。给cp的最后一个变量必须是已存在的目录名。
-p保留原先文件的所有者、组、权限和时间标志。
-r递归拷贝目录。
-s建立符号连接，替代非目录文件拷贝。
-v在拷贝前打印每个文件名。
-R同-r
--help在标准输出上打印帮助信息并退出。
--version在标准输出上打印版本信息并退出。
crontab
操作每个用户的守护程序和该执行的时间表。
语法：
crontabfile[-uuser]用指定的文件替代目前的crontab。
crontab－[-uuser]用标准输入代替当前的crontab
crontab－l[user]列出用户目前的crontab
crontab－e[user]编辑用户目前的crontab
crontab－d[user]删除用户目前的crontab
crontab－cdir指定crontab的目录
crontab文件的格式是：MHDmdcmd
M分钟（0-59）。
H小时（0-23）。
D天（1-31）。
m月（1-12）。
d一星期内的一天（0-6，0为星期日）。
cmd要运行的程序。
date
显示系统的时间和日期，root可以使用该命令来设置日期和时间。用户可以控制日期如何显示。
语法：
dateMMDDhhmmyy或date+formate
MM月（0-12）。
DD日期（1-31）。
hh小时（00-23）。
mm分钟（00-59）。
yy年（00-99）。
+format由%和紧跟其后的下列字符组成：

n插入一行。
t插入tab。
m月份（数字）。
d日期（数字）。
y年份的最后两位。
Dnn/dd/yy格式的日期。
I小时。
M分钟。
s秒。
W星期（0-6，0为星期日）
dd
拷贝一个文件
语法：

dd[-help][-version][if=file][of=file][ibs=bytes][obs=bytes][bs=bytes][cbs=bytes]

[skip=blocks][seek=blicks][conv={ascii,ebcdic,ibm,block,unblock,lcase,ucase,
swab,noerror,notrunc,sync}]
if=file以file作为源文件代替标准输入。
of=file以file作为目标文件代替标准输出。
ibs=bytes一次读bytes个byte。
obs=bytes一次写bytes个byte。
cbs=bytes一次读写bytes个byte，这个参数将改写ibs和obs的值。
skip=blocks在开头忽略blocks个ibs大小的块。
seek=blocks在开头忽略blocks个obs大小的块。
count=blocks只拷贝blocks个ibs大小的块。

conv=conversion[,conversion...]转换文件，可转换的文件格式有
ascii,ebcdic,ibm,block,unblock,lcase,ucase,swab,noerror,notrunc,sync。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
df
检查磁盘剩余空间
语法：
df[参数]
-a列出block为零的文件系统缺省是不列出的。
-i用inode的使用情况来代替block的使用情况。
-k以K为单位输出block（默认是512bytes）。
-P使用Posix格式输出。
-T输出每个文件系统的类型。
-t只输出列出在fstype中的类型的文件系统。
-x只输出不在fstype中的类型的文件系统。
-v无用，只是为了和SystemV版本的df兼容。
--help在标准输出上输出帮助信息并退出。

--version在标准输出上输出版本信息并退出。
diff
比较两个文本文件的不同
语法：
diff[参数]源文件目标文件
-a将所有文件当做文本文件来处理。
-b忽略空格造成的不同。
-B忽略空行造成的不同。
-q只报告何处不同，不报告具体信息。
-c使用纲要输出格式。
-e输出ed可以编辑的格式。
-f输出同-e类似的反序格式。
-H利用试探法加速对大文件的搜索。
-i忽略大小写的变化。
-l用pr对输出分页。
-n输出RCS格式。
-r比较目录时比较所有的子目录。
-v在标准输出上输出版本信息并退出。
du
报告磁盘空间使用情况
语法：
du[参数]
-a显示文件的总合。
-b以byte为单位输出所占空间的大小。
-c在处理完所有的参数后给出所有这些参数的总计。
-k以kilobytes为单位输出所占空间的大小。
-l统计所有文件的大小即使它已经在另一个连接中被统计过了。
-s对每个参数只显示总合。
-x－onefilesystem忽略不在本文件系统上的目录。
-D不统计命令行参数的符号链接的文件。
-L不统计符号链接的文件。
-S单独统计每个目录的大小，不包括子目录的大小。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
echo
本命令传给它的参数在标准输出上显示。
语法：
echo[参数]字符串
-a不输出新行。
-e使用echo显示下列Esc字符。
\a警告（响铃）。
\b退格。
\c在行尾不另起一行。
\f换页。
\n换行。
\r回车。
\t制表符。

\V垂直制表符。
\\反斜杠。
\nnn字符的八进制码是nnn。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
env
修改命令执行环境，不影响当前环境。也可显示当前环境。
语法：
env[参数]变量命令
-u从原环境中删除变量
-i开始一个空环境，忽略原环境的所有设置。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
file
检查文件的类型
语法：
file[参数][-fnamefile][-mmagicfile]文件名
-v在标准输出上输出版本信息并退出。
-mmagicfile指定用来代替/etc/magic的新文件。
-z检查压缩过的文件。
-c输出正在处理的magicfile的分析表格，常同-m参数联合起来使用测试新的magicfile文件。
-fnamefile从文件namefile中读取要分析的文件名列表。
-L允许符号链接。
文件名要分析的文件名。
find
搜索特定的文件
语法：
find[路径..][匹配表达式]
路径要搜索的目录。
匹配表达式要搜索的文件匹配标准或说明。
-name文件名告诉find要查找的文件。
-perm模式匹配所有符合指定数值模式值的文件，如果模式前面是“-”号，则搜索所有除这个模式以外的所有模式。
-sizen匹配大小为n个block的文件名。
-user用户名搜索所有属主为用户名的文件。
-group组名搜索所有属主为组名的文件。
-atimen搜索在n天前访问过的文件。
-mtimen搜索在n天前修改过的文件。
-exec命令对每个匹配的文件执行命令，标志{}用于指定命令执行时文件名出现的地方，命令必须终止于符号“\;”。
-print将搜索结果输出到标准输出。
finger
显示系统中用户的信息。
语法：
funger[参数]用户名
-b在输出结果中略去用户的home和shell。
-f隐藏头部几行。

-l强迫长输出。

-p不打印.plan文件。

-q显示用户的快速列表。

-s显示一个短格式。

user指定登录名、实际名和姓。

fortune

输出一段寓言或谚语

语法：

fortune

grep

在文件中搜索匹配的行并输出。

语法：

grep[-[AB]]num][-[CEFGVBchilnsvwx]][-e]正确表达式|-ffile][文件名]

-Anum输出匹配行后的num行。

-Bnum输出匹配行前的num行。

-V输出版权信息。

-ffile从文件中读入表达式。

-q不输出。

-s不输出错误信息。

gzexe

将可执行文件压缩成较小的可执行文件。

语法：

gzexe[文件名...]

-d解压缩

gzip

压缩或解压缩文件

语法：

gzip[参数]文件名。

-d解压缩。

-h显示帮助信息并退出。

-t检验压缩文件。

-V显示版权信息并退出。

-v输出压缩信息。

-##为0-9的数字，数字越小速度越快，但是压缩比越小。

-c压缩结果输出到标准输出，如不加-c参数，则gzip将压缩为.gz文件并删除源文件。

head

显示文件的前几行。

语法：

head[参数]文件

-cN输出文件的前N个字节。

-nN输出文件的前N行。

-q不输出文件名的信息。

-v输出文件名的信息。

--help在标准输出上输出帮助信息并退出。

--version在标准输出上输出版本信息并退出。

id

显示用户的系统标志，报告用户名、用户号（ID）、组名和组号（GID）。

语法：
id[参数]
-g只显示组号。
-G只显示次要组。
-u只显示用户号。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
kill
终止当前进程。
语法：
kill[信号]进程号

信号送入可选信号，缺省值为SIGTERM。其它两个常用值，一个是SIGHUP，是调制解调器通过电话挂起的设备；另一个是SIGKILL，不能被进程忽略。
-l显示可用kill传送的信号名称表。
less
按页显示文件。类似于more，但允许在文件中前后移动。
语法：
less[参数]文件名
-？输出less可以接受的命令选项。
-a在屏幕上输出最后一行之后开始查询。
-c从顶行向下全屏重写。
-C与-c同，但写之前清屏。
-e第二次到文件尾部时退出less。默认的退出方法是按“q”键。
-E第一次到文件尾后自动退出less。
-i查询时忽略大小写。
-n去掉行号。
-ofile将输出拷贝到文件。
-q不输出响铃。
-Q同-q。
-s将多个空行压缩成一个空行。
-xn每次按制表符走n格，n的默认值是8。
ln
在文件间建立链接。
语法：
ln[参数]源文件目标文件
-b为删除的文件建立备份。
-d，-F允许root建立硬连接。
-f删除目标文件。
-i在删除文件时给出提示。
-n如果目标文件是一个目录的符号链接，则代替此符号链接，而不是在这个目录中做链接。
-s用符号链接代替硬连接。
-v输出每个要连接的文件的文件名。
--help在标准输出上输出帮助信息然后退出。
--version在标准输出上输出版本信息然后退出。
ls

列出目录中的内容。
语法：
ls[参数][目录]
-A显示所有文件，包括隐藏文件（以“.”开头的文件，但不列出当前目录“.”和上级目录“..”。
-a显示所有文件，包括“.”和“..”目录。
-c按列列出文件。
-x按行列出文件。
-d只列出目录名；不列出它的内容。
-l给出文件的长列表。包括权限、所有权、大小、文件名、最后修改时间等等。
mail
阅读和发送邮件给其它的用户
语法：
mail[参数][用户名]
-？输出该命令的一个摘要。
-e检查收件箱中的邮件。
-ffile打开文件来阅读邮件，如果没有指定文件，则使用mbox文件。
-F把发送邮件保存在和第一个收到信息同名的文件中去。
-H显示邮件箱中内容的头部信息概要。
-sSubject把邮件头部中的主题栏设置为Subject。
-u阅读用户的收件箱。
man
显示具有一定格式的在线手册。
语法：
man[参数][section]查询名
-a显示匹配查询名的所有信息。
-b在输出中留空行。
-ddir把指定目录dir加到搜索路径中。
mcopy
从DOS文件系统拷贝文件或把文件拷贝到DOS文件系统。
语法：
mcopy[参数]源文件目标文件
-t文本文件传输。Mcopy将回车/换行翻译成换行。
-n当覆盖一个已存在的文件时不警告用户。
-m保存文件修改时间。
mdir
显示MS-DOS目录的内容。
语法：
mdir[-w]目录名
-w用没有文件大小和建立日期的宽格式显示目录。
mesg
选择接收或不接收其它用户发送来的信息。
语法：
mesg[y][n]

此命令用来控制接收其它用户利用talk或write向用户发出的信息，如果不想接收的话，使用mesgn就可以不接受别的用户向用户的终端发送过来的信息。

mkdir
建立新的目录
语法：
mkdir[参数]目录名
-mmode为新目录设置访问的模式。
-p如果父目录不存在，创建所有的父目录。
more
分屏显示文件。按Enter键显示下一行，按空格键显示下一屏。
语法：
more[参数][文件名]
-c按页显示文件时，清空每个窗口而非卷动，因为这要快一些。
-f统计逻辑行数而非屏幕行数。
-l略过from-feed字符。
-r强制以“^X”来显示控制字符。
-s把多行空白用一行显示。
mv
移动或更名文件和目录
语法：
nv[参数]源文件目标文件
-b为移动的文件制作备份。
-f强制覆盖已有文件。
-i在覆盖已有文件前提示用户。
-u在目标文件的时间比原文件新时不覆盖目标文件。
-v在移动文件时输出相关信息。
passwd
设置用户口令。
语法：
passwd[参数]用户名
-l屏蔽一个用户的帐号
-n设定用户只有在min天后才可以修改密码。
-s显示口令信息，如用户名、口令状态、最后改动时间。
-w在max天后用户需要修改密码。
-x设置用户密码有效的最长时间。
ps
报告进程状态。由于进程在执行中运行得很快，本报告只能反映查询瞬间的状态。
语法：
ps[参数]
-l给出长列表。
-u显示用户名和起始时间。
-j按作业格式进行输出。
-s按信号格式进行输出。
-v按虚拟内存格式显示。
-m显示内存信息。
-a显示其它用户的进程。
-x显示不带控制终端的进程。
-S增加子CPU时间和页面出错。
-c从内核的task_structune上列出命令行。

-e显示环境。
-w用宽格式输出。
-h不显示题头。
-r只显示正在运行的进程。
-n为USER和WCHAN提供数字输出。
-txx只显示受tty.xx控制的进程。
下面为所显示报告的各列及其说明：
PID进程号。
PRI进程优先级。
NILinux进程的nice值。负值意味着占用较小的CPU时间。
SIZE虚拟映像的大小，大小的计算为文本+数据+栈。
RSS驻留空间的大小。显示当前常驻内存的程序的K字节数。
WCHAN进程等待的内核时间的名称。
STAT进程的状态，用下面代码中的一个给出。
R可执行的。
S睡眠状态。
D不间断睡眠。
T停止或跟踪。
Z僵持。
W进程没有驻留页。
TT控制进程的tty名称。
PAGEIN造成从磁盘读取页出错页面号。
TRS文本驻留大小。
SWAP交换设备上的K字节数。
pwd
显示正在工作或当前的目录名。
pwd[参数]
--help在标准输出上输出帮助信息然后退出。
--version在标准输出上输出版本信息然后退出。
rcp
远程拷贝命令。
语法：
rcp[参数]主机名：文件1主机名文件2
-r如果源文件是目录，则拷贝所有子目录，目标文件也必须是目录。
-p尽量保持文件的修改时间和访问权限。
-k要求rcp获得外部许可。
-x给拷贝的数据流用DES方法加密。
reset
将终端复位。

如果由于某种原因终端在使用的过程中花掉了，这时使用reset就可以恢复。Reset调用tput函数，向终端发出复位信号。如果屏幕被Ctrl+s锁住了，可以使用Ctrl+q解除锁定。
rm
删除一个或多个文件。
语法：
rm[参数]文件或目录

-f忽略不存在的文件，并且不给用户任何提示。
-i在删除每个文件时提示用户。
-r删除目录。
-v删除每个文件时输出文件信息。
--help在标准输出上输出帮助信息然后退出。
--version在标准输出上输出版本信息然后退出。
rmdir
删除空目录。
语法：
rmdir[参数]目录列表
sdiff
产生文件1和文件2的并排比较，输出为：
texttext相同行。
text<行只存在于文件1中。
>text行只存在于文件2中。
text|text行不同。
语法：
sdiff[参数]文件1文件2
-wwidth指定输出行宽度为width个字符。
-l相同的行只输出在输出的左边。
-s不显示相同的行。
文件1输出中处于左边的文件。
文件2输出中处于右边的文件。
set
显示shell变量的值
setcolor
设置屏幕的颜色和其它的属性。
语法：
-bcolor把背景颜色设置为color。
-gcolor1color2把前景颜色设置为color1，背景颜色设置为color2。
-n把屏幕复位为默认值白色前景黑色背景。
-rcolor1color2设置屏幕的反转显示前景为color1，反转显示背景为color2。
-ocolor设置边框为color。
sync
清空Linux文件系统的缓冲区。

在系统的CPU停止工作以前必须调用该命令来将系统的缓冲区写入硬盘。Reboot和halt命令都调用该命令来清空缓冲区。Linux的优点之一就是磁盘存取速度很快，这是因为它将不用的内存都拿来当缓冲，这样系统在关机之前必须将缓冲清空，否则就会丢失数据。
su
改变用户的ID或变成root
语法：
su用户名
tail
显示文件的最后一部分。
语法：

tail[参数]文件名

tail默认显示文件列表中每个文件的后10行，如果没有文件名或文件名为“-”则其从标准输入中读取文件，如果有多个文件则其会在文件前面加上“==>文件名<==”以便区别。
-cN显示文件后部的N比特大小的部分，N后面可以跟bkm参数。
b512比特的块。
k1k的块。
m1M的字节块。
-f如果文件的大小在增长的话，tail将跟随文件的增长而显示。
-l,-nN显示文件的尾部N行。
-v一直输出“==>文件名<==”形式的文件。
--help在标准输出上显示帮助信息然后退出。
--version在标准输出上显示版本信息然后退出。
tar
从一个打包介质中拷贝文件或恢复文件。如果files是目录，则整个子目录都一起操作，参数不需在文件之前。
语法：
tar[参数]文件
-A向归档文件中追加另一个tar文件。
-c建立新的归档文件。
-d将归档文件和系统文件进行比较。
--delete从归档文件中删除文件。
-r向归档文件末尾追加文件。
-t对归档文件列目录。
-u只追加比归档文件中的文件更新的文件。
-x从归档文件中解开文件。
-bN块的大小为N乘以512比特。
-B为兼容BSD而设。
-CDIR将目录名改为DIR。
--checkpoint在读归档文件时输出文件名。
-v处理时显示文件名。
tcsh

是类似bash的命令解释程序，是通用Cshell的扩展。与bash一样，tcsh自己是完整的程序编辑语言，具体信息请查看mannpages。
tee
从标准输入中读取向标准输出或其它文件中写入。
语法：
tee[参数]文件
-a把输出附加到文件中，而不是覆盖它们。
-i忽略所有的中断。
-u使输出没有缓冲。
--help在标准输出上输出帮助信息然后退出。
--version在标准输出上输出版本信息然后退出。
test

最常见于if语句和while。If语句和while是shell的控制结构。在Bourneshell中编程使用。当要检测的结果为真时，test返回0，退出。
语法：
test[表达式]
-b文件名如果文件存在且为块文件则为真。
-c文件名如果文件存在且为字符文件则为真。
-d文件名如果文件存在且为目录则为真。
-e文件名如果文件存在则为真。
-f文件名如果文件存在且为普通文件则为真。
-g文件名如果文件存在且为设置组则为真。
-k文件名如果文件是真实文件则为真。
-L文件名如果文件存在且为符号连接文件则为真。
-p文件名如果文件存在且为管道文件则为真。
-r文件名如果文件存在且为可读文件则为真。
-s文件名如果文件存在且比上一个文件大则为真。
-S文件名如果文件存在且为socket文件则为真。
-t[fd]如果fd在一个端口上打开则为真。
-u文件名如果文件存在且为其设置用户ID的位为1则为真。
-w文件名如果文件存在且可写则为真。
-x文件名如果文件存在且可执行则为真。
-O文件名如果文件存在且其属主为有效用户则为真。
-G文件名如果文件存在且属主为有效组则为真。
文件1－nt文件2如果文件1比文件2新则为真。
文件1－ot文件2如果文件1比文件2旧则为真。
文件1－ef文件2如果文件1比文件2在相同的设备上并且有相同的inode则为真。
-z字符串如果字符串长度为0则为真。
-n字符串如果字符串长度不为0则为真。
字符串1=字符串2如果两个字符串相等则为真。
字符串1！=字符串2如果两个字符串不相等则为真。
！表达式如果表达式为假则为真。
表达式1－a表达式2：表达式1和表达式2都为真则返回真。
表达式1-o表达式2：表达式1或表达式2只要有一个为真就返回真。
time
确定程序执行多长时间。
语法：
time命令
time报告了三个不同时间：
real从开始命令起全部占用的时间，有时被称为“挂钟”时间。
user花费CPU外部系统时间的时间总和。
sys花费内核时间的总和，就是花费在满足系统需要的时间的总和。
所有CPU时间是user+sys时间的总和，这与real时间的不同之处在于real包含CPU花费在其它任务上的时间。
top
显示系统的最高进程。

该命令可以即时显示当前系统占用CPU时间的进程，它同时提供一个交互的界面让用户可以观察系统进程情况，并按照CPU使用情况、占用内存大小、运行时间来对进程排序，这是系统管理的一项必不可少的工具。
语法：
top[参数]
d设定刷新屏幕的时间间隔。
q此选项使top刷新时间为零，如果超级用户使用这个选项的话，则top以最高优先级运行。
S设定累积模式，即对外界输入不作任何反应。
i忽略死进程和僵尸进程。
c显示命令行。
在运行top时按下“h”键就可以观看top交互命令的帮助。
tr

将文件中的字符从一种形式翻译或变换成另一种形式。例如，可以用tr将所有制表符该为空格。本命令用相对简单的结构能够做比较有力的字符处理。
语法：
tr[参数]源字符串目标字符串

-c通常tr命令用目标串中的字符替换源串字符，在输出部分显示文件发生替换的原始内容。本选项限定指定的源串字符输出，并有效地附加字符到目标串。
-d删除源串中指定的字符。
-s去除目标串中指定的在输出时产生的重复字符，在输出中只剩下一个重复的字符。
tty
报告当前登录的终端设备名或检测标准输入是否是终端。
语法：
tty－s
-s本选项测试标准输入是否是终端，不产生输出。若标准输入是终端，则结果为0，否则为1。
mount unmount
mount安装一个文件系统。
umount卸下一个文件系统。
语法：
mount－a[参数][文件系统类型]
-V输出版权信息。
-h输出帮助信息。
-v校验模式。
-amount在文件/etc/fstabb中定义的所有文件系统。
-F同-a连用，并行mount所有的文件系统。
-f与-v连用，不做真实mount过程只是检查输出。
-n安装文件系统后不写入/etc/mtab。
-r将文件系统安装为只读模式。
-w将文件系统安装成读写模式。

-tvfstype指定文件系统的类型。如：minix，ext，ext2，xiafs，hpfs，msdos，umsdos，vfat，nfs，

iso9660，smbfs等等
-o设定mount的一些参数。
async文件系统必须被异步写入。
atime每次访问时将i节点的访问时间更新。
auto自动选择文件系统的类型。
defaults使用默认选项suid，dev，exec，auto，nouser与async等。
dev解释文件系统上的特殊设备文件。
exec允许运行文件系统上的可执行文件。
noatime每次访问时不将i节点的访问时间更新。
noauto禁止自动选择文件系统类型。
nodev禁止解释文件系统上的特殊设备文件。
noexec禁止运行文件系统上的可执行文件。
nosuid禁止设定用户ID位和设定组位起作用。
nouser禁止非root用户使用。
remount重新mount一个已mount的文件系统。
ro将文件系统mount为只读模式。
rw将文件系统mount为读写模式。
suid允许设定用户ID位和设定组位起作用。
sync文件系统必须被同步写入。
user允许非超级用户使用。
uname
显示系统信息。
语法：
uname[参数]
-m显示计算机硬件种类。
-n显示计算机的网络节点名。
-r显示操作系统发布版本。
-s显示操作系统名称。
-v显示操作系统发行的版本。
-a显示上面所有的信息。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
wall
向所有的用户发布信息。
语法：
wall文件名

wall将给定文件名输出到所有登录用户的屏幕上，如果不给出文件名则以标准输入作为输入，以Ctrl-D作为文件结束符。只有root才能向那些设为屏蔽信息状态的用户的屏幕上输出信息。
wc
统计文件中字符、字和行的数量。
语法：
wc[参数]文件
-c只输出字节数。
-w只输出单词数。
-l只输出行数。

--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
who
查看当前系统上的用户和其它用户及登录信息。
语法:
who[参数]utmp式的文件
utmp式的文件这是一个二者挑一的文件，读取它可以获得登录信息，通常是/var/log/wtmp。
-m同whoami和“whoami”
-q只输出用户的登录名和正在使用的用户数目。
-i输出用户发呆时间，“.”表示用户没有发呆，“old”表明用户发呆超过24小时。
-H输出报头。
-w,-T输出用户的信息屏蔽状态。
+可写。
-不可写。
? 不能发现终端设备。
--help在标准输出上输出帮助信息并退出。
--version在标准输出上输出版本信息并退出。
write
向另一个用户发送信息。
如果用户不是root，则不能对设定messagen的用户发送信息，输入完毕后，使用Ctrl-D结束。
zcat
解压缩一个文件（以.z结尾）内容，而不改变压缩文件本身。
语法:
zat文件.z

Linux启动过程综述

我们一起在风雨中前行

Linux启动过程综述 作者:杨沙洲

内容:

五.init进程和inittab引导指令

六.rc启动脚本

七.getty和login

八.bash

附：XDM方式登录 作者:杨沙洲

本文以Redhat 6.0 Linux 2.2.19 for Alpha/AXP为平台，描述了从开机到登录的Linux 启动全过程。该文对i386平台同样适用。

一. Bootloader

在Alpha/AXP平台上引导Linux通常有两种方法，一种是由MILO及其他类似的引导程序引 导，另一种是由Firmware直接引导。MILO功能与i386平台的LILO相近，但内置有基本的磁盘驱动程序（如IDE、SCSI等），以及常见的文件系统驱动程序（如ext2，iso9660等）， firmware有ARC、SRM两种形式，ARC具有类BIOS界面，甚至还有多重引导的设置；而SRM则具有功能强大的命令行界面，用户可以在控制台上使用boot等命令引导系统。ARC有分区 （Partition）的概念，因此可以访问到分区的首扇区；而SRM只能将控制转给磁盘的首扇区。两种firmware都可以通过引导MILO来引导Linux，也可以直接引导Linux的引导代码。

“arch/alpha/boot”下就是制作Linux Bootloader的文件。“head.S”文件提供了对OSF PAL/1的调用入口，它将被编译后置于引导扇区（ARC的分区首扇区或SRM的磁盘0扇区）， 得到控制后初始化一些数据结构，再将控制转给“main.c”中的start_kernel()，start_kernel()向控制台输出一些提示，调用pal_init()初始化PAL代码，调用openboot() 打开引导设备（通过读取Firmware环境），调用load()将核心代码加载到START_ADDR（见“include/asm-alpha/system.h”），再将Firmware中的核心引导参数加载到ZERO_PAGE(0) 中，最后调用runkernel()将控制转给0x100000的kernel，bootloader部分结束。

“arch/alpha/boot/bootp.c”以“main.c”为基础，可代替“main.c”与“head.S”生成用于BOOTP协议网络引导的Bootloader。

Bootloader中使用的所有“srm_”函数在“arch/alpha/lib/”中定义。

以上这种Boot方式是一种最简单的方式，即不需其他工具就能引导Kernel，前提是按照 Makefile的指导，生成bootimage文件，内含以上提到的bootloader以及vmlinux，然后将bootimage写入自磁盘引导扇区始的位置中。

当采用MILO这样的引导程序来引导Linux时，不需要上面所说的Bootloader，而只需要 vmlinux或vmlinux.gz，引导程序会主动解压加载内核到0x1000（小内核）或0x100000（大内核），并直接进入内核引导部分，即本文的第二节。

对于I386平台

i386系统中一般都有BIOS做最初的引导工作，那就是将四个主分区表中的第一个可引导 分区的第一个扇区加载到实模式地址0x7c00上，然后将控制转交给它。

在“arch/i386/boot”目录下，bootsect.S是生成引导扇区的汇编源码，它首先将自己 拷贝到0x90000上，然后将紧接其后的setup部分（第二扇区）拷贝到0x90200，将真正的内核代码拷贝到0x100000。以上这些拷贝动作都是以bootsect.S、setup.S以及vmlinux在磁盘上 连续存放为前提的，也就是说，我们的bzImage文件或者zImage文件是按照bootsect，setup，vmlinux这样的顺序组织，并存放于始于引导分区的首扇区的连续磁盘扇区之中。

bootsect.S完成加载动作后，就直接跳转到0x90200，这里正是setup.S的程序入口。 setup.S的主要功能就是将系统参数（包括内存、磁盘等，由BIOS返回）拷贝到
0x90000-0x901FF内存中，这个地方正是bootsect.S存放的地方，这时它将被系统参数覆盖。 以后这些参数将由保护模式下的代码来读取。

除此之外，setup.S还将video.S中的代码包含进来，检测和设置显示器和显示模式。最 后，setup.S将系统转换到保护模式，并跳转到0x100000（对于bzImage格式的大内核是
0x100000，对于zImage格式的是0x1000）的内核引导代码，Bootloader过程结束。
对于2.4.x版内核
没有什么变化。

二.Kernel引导入口

在arch/alpha/vmlinux.lds的链接脚本控制下，链接程序将vmlinux的入口置于
"arch/alpha/kernel/head.S"中的__start上，因此当Bootloader跳转到0x100000时，
__start处的代码开始执行。__start的代码很简单，只需要设置一下全局变量，然后就跳转 到
start_kernel去了。start_kernel()是"init/main.c"中的asmlinkage函数，至此，启
动过程转入体系结构无关的通用C代码中。
对于I386平台
在i386体系结构中，因为i386本身的问题，在"arch/alpha/kernel/head.S"中需要更多的设置，但最终也是通过call
SYMBOL_NAME(start_kernel)转到start_kernel()这个体系结构无关的函数中去执行了。

所不同的是，在i386系统中，当内核以bzImage的形式压缩，即大内核方式 （__BIG_KERNEL__）压缩时就需要预先处理bootsect.S和setup.S，按照大核模式使用$(CPP)
处理生成bbootsect.S和bsetup.S，然后再编译生成相应的.o文件，并使用
"arch/i386/boot/compressed/build.c"生成的build工具，将实际的内核（未压缩的，含
kernel中的head.S代码）与"arch/i386/boot/compressed"下的head.S和misc.c合成到一起，其中的
head.S代替了"arch/i386/kernel/head.S"的位置，由Bootloader引导执行
（startup_32入口），然后它调用misc.c中定义的decompress_kernel()函数，使用 "lib/inflate.c"中定义的gunzip()将内核解压到0x100000，再转到其上执行
"arch/i386/kernel/head.S"中的startup_32代码。
对于2.4.x版内核
没有变化。

三.核心数据结构初始化--内核引导第一部分

start_kernel()中调用了一系列初始化函数，以完成kernel本身的设置。 这些动作有的是公共的，有的则是需要配置的才会执行的。

在start_kernel()函数中，
输出Linux版本信息（printk(linux_banner)）
设置与体系结构相关的环境（setup_arch()）

页表结构初始化（paging_init()）
使用"arch/alpha/kernel/entry.S"中的入口点设置系统自陷入口（trap_init()）
使用alpha_mv结构和entry.S入口初始化系统IRQ（init_IRQ()）
核心进程调度器初始化（包括初始化几个缺省的Bottom-half，sched_init()）
时间、定时器初始化（包括读取CMOS时钟、估测主频、初始化定时器中断等，time_init()）
提取并分析核心启动参数（从环境变量中读取参数，设置相应标志位等待处理，（parse_options()）
控制台初始化（为输出信息而先于PCI初始化，console_init()）
剖析器数据结构初始化（prof_buffer和prof_len变量）
核心Cache初始化（描述Cache信息的Cache，kmem_cache_init()）
延迟校准（获得时钟jiffies与CPU主频ticks的延迟，calibrate_delay()）
内存初始化（设置内存上下界和页表项初始值，mem_init()）
创建和设置内部及通用cache（"slab_cache"，kmem_cache_sizes_init()）
创建uid taskcount SLAB cache（"uid_cache"，uidcache_init()）
创建文件cache（"files_cache"，filescache_init()）
创建目录cache（"dentry_cache"，dcache_init()）
创建与虚存相关的cache（"vm_area_struct"，"mm_struct"，vma_init()）

块设备读写缓冲区初始化（同时创建"buffer_head"cache用户加速访问，buffer_init()）
创建页cache（内存页hash表初始化，page_cache_init()）
创建信号队列cache（"signal_queue"，signals_init()）
初始化内存inode表（inode_init()）
创建内存文件描述符表（"filp_cache"，file_table_init()）
检查体系结构漏洞（对于alpha，此函数为空，check_bugs()）
SMP机器其余CPU（除当前引导CPU）初始化（对于没有配置SMP的内核，此函数为空，smp_init()）
启动init过程（创建第一个核心线程，调用init()函数，原执行序列调用cpu_idle() 等待调度，init()）
至此start_kernel()结束，基本的核心环境已经建立起来了。
对于I386平台
i386平台上的内核启动过程与此基本相同，所不同的主要是实现方式。
对于2.4.x版内核
2.4.x中变化比较大，但基本过程没变，变动的是各个数据结构的具体实现，比如Cache。
四.外设初始化--内核引导第二部分
init()函数作为核心线程，首先锁定内核（仅对SMP机器有效），然后调用 do_basic_setup()完成外设及其驱动程序的加载和初始化。过程如下：

总线初始化（比如pci_init()）
网络初始化（初始化网络数据结构，包括sk_init()、skb_init()和proto_init()三部分，在proto_init()中，将调用protocols结构中包含的所有协议的初始化过程，sock_init()）

创建bdflush核心线程（bdflush()过程常驻核心空间，由核心唤醒来清理被写过的内存缓冲区，当bdflush()由kernel_thread()启动后，它将自己命名为kflushd）

创建kupdate核心线程（kupdate()过程常驻核心空间，由核心按时调度执行，将内存缓冲区中的信息更新到磁盘中，更新的内容包括超级块和inode表）

设置并启动核心调页线程kswapd（为了防止kswapd启动时将版本信息输出到其他信息中间，核心线调用kswapd_setup()设置kswapd运行所要求的环境，然后再创建

kswapd核心线程）
创建事件管理核心线程（start_context_thread()函数启动context_thread()过程，并重命名为keventd）

设备初始化（包括并口parport_init()、字符设备chr_dev_init()、块设备 blk_dev_init()、SCSI设备scsi_dev_init()、网络设备net_dev_init()、磁盘初始化及分区检查等等，device_setup()）

执行文件格式设置（binfmt_setup()）
启动任何使用__initcall标识的函数（方便核心开发者添加启动函数，do_initcalls()）
文件系统初始化（filesystem_setup()）
安装root文件系统（mount_root()）
至此do_basic_setup()函数返回init()，在释放启动内存段（free_initmem()）并给内核解锁以后，init()打开/dev/console设备，重定向stdin、stdout和stderr到控制台，最后，搜索文件系统中的init程序（或者由init=命令行参数指定的程序），并使用
execve()系统调用加载执行init程序。

init()函数到此结束，内核的引导部分也到此结束了，这个由start_kernel()创建的第一个线程已经成为一个用户模式下的进程了。此时系统中存在着六个运行实体：

start_kernel()本身所在的执行体，这其实是一个"手工"创建的线程，它在创建了init()线程以后就进入cpu_idle()循环了，它不会在进程（线程）列表中出现

init线程，由start_kernel()创建，当前处于用户态，加载了init程序
kflushd核心线程，由init线程创建，在核心态运行bdflush()函数
kupdate核心线程，由init线程创建，在核心态运行kupdate()函数
kswapd核心线程，由init线程创建，在核心态运行kswapd()函数
keventd核心线程，由init线程创建，在核心态运行context_thread()函数
对于I386平台
基本相同。
对于2.4.x版内核
这一部分的启动过程在2.4.x内核中简化了不少，缺省的独立初始化过程只剩下网络 （sock_init()）和创建事件管理核心线程，而其他所需要的初始化都使用__initcall()宏
包含在do_initcalls()函数中启动执行。

五.init进程和inittab引导指令

init进程是系统所有进程的起点，内核在完成核内引导以后，即在本线程（进程）空 间内加载init程序，它的进程号是1。
init程序需要读取/etc/inittab文件作为其行为指针，inittab是以行为单位的描述性（非执行性）文本，每一个指令行都具有以下格式：

id:runlevel:action:process其中id为入口标识符，runlevel为运行级别，action为动作代号，process为具体的执行程序。

id一般要求4个字符以内，对于getty或其他login程序项，要求id与tty的编号相同，否则getty程序将不能正常工作。

runlevel是init所处于的运行级别的标识，一般使用0－6以及S或s。0、1、6运行级别被系统保留，0作为shutdown动作，1作为重启至单用户模式，6为重启；S和s意义相同，表示单用户模式，且无需inittab文件，因此也不在inittab中出现，实际上，进入单用户模式时，init直接在控制台（/dev/console）上运行/sbin/sulogin。

在一般的系统实现中，都使用了2、3、4、5几个级别，在Redhat系统中，2表示无NFS支持的多用户模式，3表示完全多用户模式（也是最常用的级别），4保留给用户自定义，5表示XDM图形登录方式。7－9级别也是可以使用的，传统的Unix系统没有定义这几个级别。runlevel可以是并列的多个值，以匹配多个运行级别，对大多数action来说，仅当runlevel与当前运行级别匹配成功才会执行。

initdefault是一个特殊的action值，用于标识缺省的启动级别；当init由核心激活 以后，它将读取inittab中的initdefault项，取得其中的runlevel，并作为当前的运行级
别。如果没有inittab文件，或者其中没有initdefault项，init将在控制台上请求输入 runlevel。

sysinit、boot、bootwait等action将在系统启动时无条件运行，而忽略其中的runlevel，其余的action（不含initdefault）都与某个runlevel相关。各个action的定义在inittab的man手册中有详细的描述。

在Redhat系统中，一般情况下inittab都会有如下几项：
id:3:initdefault:
#表示当前缺省运行级别为3--完全多任务模式；
si::sysinit:/etc/rc.d/rc.sysinit
#启动时自动执行/etc/rc.d/rc.sysinit脚本
l3:3:wait:/etc/rc.d/rc 3
#当运行级别为3时，以3为参数运行/etc/rc.d/rc脚本，init将等待其返回
0:12345:respawn:/sbin/mingetty tty0
#在1－5各个级别上以tty0为参数执行/sbin/mingetty程序，打开tty0终端用于
#用户登录，如果进程退出则再次运行mingetty程序
x:5:respawn:/usr/bin/X11/xdm -nodaemon
#在5级别上运行xdm程序，提供xdm图形方式登录界面，并在退出时重新执行

六.rc启动脚本

上一节已经提到init进程将启动运行rc脚本，这一节将介绍rc脚本具体的工作。

一般情况下，rc启动脚本都位于/etc/rc.d目录下，rc.sysinit中最常见的动作就是激活交换分区，检查磁盘，加载硬件模块，这些动作无论哪个运行级别都是需要优先执行的。仅当rc.sysinit执行完以后

init才会执行其他的boot或bootwait动作。

如果没有其他boot、bootwait动作，在运行级别3下，/etc/rc.d/rc将会得到执行，命令行参数为3，即执行/etc/rc.d/rc3.d/目录下的所有文件。rc3.d下的文件都是指向/etc/rc.d/init.d/目录下各个Shell脚本的符号连接，而这些脚本一般能接受start、stop、restart、status等参数。rc脚本以start参数启动所有以S开头的脚本，在此之前，如果相应的脚本也存在K打头的链接，而且已经处于运行态了（以/var/lock/subsys/下的文件作为标志），则将首先启动K开头的脚本，以stop作为参数停止这些已经启动了的服务，然后再重新运行。显然，这样做的直接目的就是当init改变运行级别时，所有相关的服务都将重启，即使是同一个级别。

rc程序执行完毕后，系统环境已经设置好了，下面就该用户登录系统了。

七.getty和login

在rc返回后，init将得到控制，并启动mingetty（见第五节）。mingetty是getty的简化，不能处理串口操作。getty的功能一般包括：

打开终端线，并设置模式
输出登录界面及提示，接受用户名的输入
以该用户名作为login的参数，加载login程序
缺省的登录提示记录在/etc/issue文件中，但每次启动，一般都会由rc.local脚本根据系统环境重新生成。
注：用于远程登录的提示信息位于/etc/issue.net中。
login程序在getty的同一个进程空间中运行，接受getty传来的用户名参数作为登录的用户名。

如果用户名不是root，且存在/etc/nologin文件，login将输出nologin文件的内容，然后退出。这通常用来系统维护时防止非root用户登录。

只有/etc/securetty中登记了的终端才允许root用户登录，如果不存在这个文件，则root可以在任何终端上登录。/etc/usertty文件用于对用户作出附加访问限制，如果不存在这个文件，则没有其他限制。

当用户登录通过了这些检查后，login将搜索/etc/passwd文件（必要时搜索 /etc/shadow文件）用于匹配密码、设置主目录和加载shell。如果没有指定主目录，将默认为根目录；如果没有指定shell，将默认为/bin/sh。在将控制转交给shell以前，
getty将输出/var/log/lastlog中记录的上次登录系统的信息，然后检查用户是否有新邮件（/usr/spool/mail/{username}）。在设置好shell的uid、gid，以及TERM，PATH
等环境变量以后，进程加载shell，login的任务也就完成了。

八.bash

运行级别3下的用户login以后，将启动一个用户指定的shell，以下以/bin/bash为例继续我们的启动过程。

bash是Bourne Shell的GNU扩展，除了继承了sh的所有特点以外，还增加了很多特性和功能。由login启动的bash是作为一个登录shell启动的，它继承了getty设置的TERM、PATH等环境变量，其中PATH对于普通用户为"/bin:/usr/bin:/usr/local/bin"，对于root
为"/sbin:/bin:/usr/sbin:/usr/bin"。作为登录shell，它将首先寻找/etc/profile
脚本文件，并执行它；然后如果存在~/.bash_profile，则执行它，否则执行 ~/.bash_login，如果该文件也不存在，则执行~/.profile文件。然后bash将作为一个交互式shell执行~/.bashrc文件（如果存在的话），很多系统中，~/.bashrc都将启动
/etc/bashrc作为系统范围内的配置文件。

当显示出命令行提示符的时候，整个启动过程就结束了。此时的系统，运行着内核，运行着几个核心线程，运行着init进程，运行着一批由rc启动脚本激活的守护进程（如
inetd等），运行着一个bash作为用户的命令解释器。

附：XDM方式登录

如果缺省运行级别设为5，则系统中不光有1-6个getty监听着文本终端，还有启动了一个XDM的图形登录窗口。登录过程和文本方式差不多，也需要提供用户名和口令，XDM
的配置文件缺省为/usr/X11R6/lib/X11/xdm/xdm-config文件，其中指定
了 /usr/X11R6/lib/X11/xdm/xsession作为XDM的会话描述脚本。登录成功后，XDM将执行这个脚本以运行一个会话管理器，比如gnome-session等。

除了XDM以外，不同的窗口管理系统（如KDE和GNOME）都提供了一个XDM的替代品，如gdm和kdm，这些程序的功能和XDM都差不多。

LINUX文件格式简介

我们一起在风雨中前行

LINUX文件格式简介

⊙ *.001, *.002 ....

大致上有以下几种可能，必须先分辨是哪种，再用对应的程序解：

1. ARJ 可以指定而造出 .001, .002 的文件
用 arj 解
2. DOS backup 造出的文件是 .001 .002
用 restore 解
3. AIN 分片压缩的档名是 .AIN, .001, .002
用 AIN 解

⊙ *.aif
MAC 上常用的音效档。PC上可用 plany 播放
⊙ *.arc
压缩档，可用 pkunpak 解压及 pak.exe 来解压；
⊙ *.arj, *.a01, *.a02 ...

这是一种压缩档，用arj 可以压缩并解压。
*.a01 等是使用arj 的分片功能时分出来的文件。
⊙ *.au
SUN 的音效档，在PC 上可用 plany 放出
⊙ *.avi
avi 是microsoft 定的一种动画档，其标准为影像加声音。
Win95 下可直接执行媒体播放程序播放。
⊙ *.bat
DOS 的 batch档，其内容为一般的文字档
⊙ *.bmp (Windows Bitmap Format)

图形档，大部份秀图程序都有支持

⊙ *.cap
画面狩猎者的输出格式，可以在原版画面狩猎者的说明书
中找到完整说明。
⊙ *.cdr (Corel DRAW)

Corel DRAW图形档
⊙ *.cmf (Creative Music File)

Creative 公司所使用类似MIDI的格式。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/cmfved10.zip
⊙ *.com
DOS 的可执行档
⊙ *.cru (crush)

压缩格式。
⊙ *.dbf (dBase file)

可到 simtel 的 dbase 目录找到相关软体。
⊙ *.dcf (Disk Copy Fast)
Disk Copy Fast造出的Image File！可用dcf来解！！
⊙ *.ddi
diskdupe 做出来的 disk image 档
可用 x2file 来解。
⊙ *.dib
图形格式，类似 *.bmp，通常可以处理 *.bmp 的程序都可以用来处理
*.dib。
⊙ *.dl
一种动画格式，可以用 dl-view 来看。

ftp://softu1.ncu.edu.tw/pub5/News/oldexe/dl-view.exe
⊙ *.dsk
diskrw 造出来的 disk image 档。
ftp://ftp.fcu.edu.tw/pub4/CDROM/simtel2/disc2/diskutil/dskrw237.zip

⊙ *.dxf (Autodesk Drawing
Exchange Format)
AutoCAD 2-d 图形档
⊙ *.eps
Encapsulated PostScriptd
⊙ *.exe
DOS 的可执行档
⊙ *.F
一种压缩格式!
ftp://ftp.edu.tw/USENET/comp.sources.misc/volume35/freeze
⊙ *.fli, *.flc
一种动画格式，*.fli 规格是小於 320*200*256色的，*.flc为
fli 的扩充规格，在dos 下可以用 aaplay , display 播放

⊙ *.gif (Graphics Interchange Format)

CompuServe 定图形档，大部份秀图程序都有支持
⊙ *.gl
一种动画格式，可以用 grasp4 来看。
ftp://ftp.edu.tw/PC/garbo/animation/grasp4.zoo
⊙ *.gz
压缩档，在unix 上可用 "gzip -d 档名" 解开
ftp://ftp.edu.tw/Chinese/ifcss/software/dos/utils/gzip124.exe(for
dos)
⊙ *.ha
压缩档。DOS 下用 HA 来压缩/解压。
ftp://ftp.edu.tw/PC/garbo/arcers/ha098.zip
⊙ *.hap
压缩档，压缩率稍比 zip 好一点，目前只有 dos 的程序。
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/hap305bp.zip
⊙ *.hpk
压缩档。DOS下用 hpack 来压缩/解压。
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/hpack78.zip
⊙ *.hqx
.hqx是MAC 上一种编码格式，主要是把8bits 的文件编为7bits，以利
传输。在PC 上可以用BINHEX 解开。
ftp://ftp.edu.tw/PC/Simtel/msdos/mac/binhex13.zip
⊙ *.html, *.htm (Hyper Text
Markup Language)
WWW 用的标准格式，在PC 上可以利用底下几个程序来读
ftp://ftp.edu.tw/WWW/client/doslynx(for dos, 只有文字，没有图形)

ftp://ftp.edu.tw/WWW/netscape/navigator/2.02/windows/n16e202.exe

ftp://ftp.edu.tw/WWW/netscape/navigator/2.02/windows/n32e202.exe

⊙ *.ico

Windows Icon files

⊙ *.iff

图形档，Amiga Images, Deluxe Paint 使用

⊙ *.img

① hd-copy 做出来的 disk image 档。

ftp://ftp.cis.nctu.edu.tw/Msdos/diskutil/hd-cp20a.zip

② 图形档 (GEM Images)

⊙ *.jis

JIS (ジ□ス) = Japanese Industrial Standard 日本工业规格

JIS 在这指的应该是日本用的一种文字内码，一般的 *.jis

都是文字档，可以用南极星来编修。

ftp://ftp.edu.tw/PC/Simtel/msdos/editor/njj30*

⊙ *.jpg

图形档(JFIF)，可以用qpeg,display 来看

⊙ *.lbm

图形档， Deluxe Paint使用，编码方式类似 *.iff

⊙ *.lzh

压缩档，可用 lha 压缩及解压

ftp://ftp.edu.tw/PC/garbo/arcers/lha255e.exe

⊙ *.mag

日本常用的一种图形档。 PC 上可以用 gv,display 来看

ftp://ftp.edu.tw/PC/garbo/graphics/disp189a.zip

ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/gv057.zip

⊙ *.mid

midi 音效档

⊙ *.mod

类似midi 音效档。

ftp://ftp.edu.tw/PC/Simtel/msdos/sound/mp219b.zip

⊙ *.mov

Quick Time 的文件，一种动画格式，为Apple 公司规画，在PC 上

可以找 Quick Time for Windows (qtw) 来看!

ftp://ftp.tem.nctu.edu.tw/Windows/multimed/qtw203.exe

ftp://ftp.cis.nctu.edu.tw/OS/Windows/95/multimedia/graphic/qt32inst.exe

⊙ *.mp2, *.mpa (Mpeg Audio Layer

2)

一般可以放mpeg 文件的player 大多可以播放。

Dos:

ftp://ftp.edu.tw/PC/Simtel/msdos/sound/ampeg43.zip

(可在.wav,.pcm<->.mp2转换)

Windows:

ftp://ftp.cis.nctu.edu.tw/Windows/3.x/graphics/softpeg13.zip

ftp://ftp.cis.nctu.edu.tw/Windows/95/multimedia/graphic/mpw32-5n.zip

⊙ *.mp3 (Mpeg Audio Layer 3)

Windows:
ftp://ftp.thu.edu.tw/upload/win95/winplay3.zip
⊙ *.mpg
一种动画格式，包含声音及影像，包含声音的方法有两种，一种是
直接放在 .mpg里，一种是另外用 .wav档储存。
ftp://ftp.edu.tw/mirror/Cica/win3/desktop/vmpeg12a.zip(For Dos)

Windows:
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/softpeg2.zip
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/xing21.zip
⊙ *.msl
一种动画格式，内含声音影像。
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/playmsl.exe
(Dos/V)
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/Maslv41.lzh
(Win3.1)
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/Masl32.lzh
(Win95)
⊙ *.pak
压缩档, 由 pak.exe 产生、解压。
ftp://ftp.edu.tw/PC/garbo/arcers/pak251.exe
⊙ *.pcc
图形档， 参见 *.pcx
⊙ *.pcd
图形档(Kodak PhotoCD)
⊙ *.pcx (ZSoft Paint)

图形档，大部份秀图程序都有支持
⊙ *.pdf (Portable Document
Format)
Adobe 制订的文件档。可用 acroread.exe 来读取。
ftp://ftp.tku.edu.tw/pc/winsock/WWW/Plug-in/AcrobatAmber
⊙ *.pic
图形档 (Lotus 1-2-3 Pictures)
⊙ *.png (Portable Network
Graphics)
图形档。为取代 GIF 而制订的图形档格式。
QPV, display, gwswin 等软件都有支持。
⊙ *.ppm (Portable Pixmap)

图形档。
⊙ *.ps
post script，一种page description language，如果你使用的列表机

有支持PS 格式的话，直接列印就好了。不然PC 上可以抓ghostview,
ghostscript 等东西来用。
ftp://ftp.edu.tw/PC/Simtel/msdos/postscrp/gs*for dos
ftp://ftp.cis.nctu.edu.tw/OS/Windows/misc/postscript/*for win
ftp://ftp.edu.tw/Windows/Windows95/Cica/miscutil/rops3244.zip
⊙ *.psd
PhotoShop 标准档。Shareware 中 Paint Shop Pro 可以读非
Multi-layer的*.psd。
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/psp311.zip
⊙ *.rar
压缩档。可以用 rar 来压缩解压。
ftp://ftp.cis.nctu.edu.tw//Msdos/archiver/rar/rar200/rar200.exe

⊙ *.ras (Sun Raster files)
图形档
⊙ *.raw
图形档(Raw GrayScale)
⊙ *.rcp
一种 Midi 音效档
⊙ *.rle
压缩过的 *.bmp(Run Length Compressed)
⊙ *.rtf
Rich Text Format
⊙ *.s3m
S3M 是由 Future Crew 所设计的一种音乐档格式，可用 iplay 来播放。
ftp://ftp.edu.tw/PC/garbo/sound/iplay121.zip
⊙ *.sdn (Software Distribution
Network 组织)
以前是采用 pak.exe 来压缩, 自 1994 以後改用 arj.exe 来压。
⊙ *.sit
Mac 上的 Stuff It 压出来的东西，压缩采用的方法是把文件集中起
来，然後用 RLE, Huffman, LZW 等方法压缩起来。底下是两个 PC
上的解压程序。
ftp://ftp.edu.tw/PC/garbo/arcers/unsit30.zip
ftp://ftp.edu.tw/PC/garbo/arcers/unstufit.zoo
⊙ *.snd
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.sqz
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/sqz1083e.exe
⊙ *.tar
为了备份方便而把所有文件放在一起的文件，在unix 上可直接下指
令`tar -xf 档名` 来解开
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/tar4dos.zip
⊙ *.tar.gz
先解开 *.gz 再把解开的 *.tar 解开。

如果是用 GNU 的tar 的话可直接用 "tar -zxvf *.tgz" 来解开。
⊙ *.td0
*.td0 是由 teledisk 造出来的disk image 档。
ftp://ftp.tem.nctu.edu.tw/Msdos/arcers/teled215.zip
⊙ *.tga (Truevision (Targa)
File Format)
图形档，svga, vpic, cshow, qpeg 都有支持
⊙ *.tgz
.tgz就是.tar.gz，先解 *.gz 再把解开的 *.tar 解开。
如果是用 GNU 的tar 的话可直接用 "tar -zxvf *.tgz" 来解开。
⊙ *.tif, *.tiff (Tag Image
File Format)
图形档， cshow、display 及一些影像绘图软件都有支持。
⊙ *.ttf
Windows True Type Font
⊙ *.u01 *.u02 .....
这是 unpack 造出来的，用 pack 可以合起来。
unpack 是把一个大文件分成数个小档的程序。
ftp://ftp.edu.tw/Chinese/Taiwan-netinfo/SEEDNET/success/apart/pack.exe

⊙ *.uc2
压缩档(UltraCompressor II)。PC 上用 uc2 压缩/解压。
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/uc2r3.exe
⊙ *.uue, *.uud
可能是由 uuencode 造出的文件，在unix 上可用 uudecode
来解码，在dos 上可以抓底下的程序来用。
ftp://ftp.edu.tw/PC/garbo/decode/uuexe651.zip
⊙ *.voc
一种音效档，可用 plany 放。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.wav
一种音效档，可用plany 放出。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.win
图形档，类似 *.tga
⊙ *.wmf
Windows Metafiles
⊙ *.xbm (X BitMap)

ftp://ftp.edu.tw/PC/garbo/graphics/disp189a.zip
⊙ *.xmi
音效档。
ftp://ftp.cis.nctu.edu.tw/Ultrasound/submit/xmi2mid.zip

⊙ *.z
参见 *.gz
⊙ *.Z
压缩档，在unix 上用 uncompress 解开
⊙ *.zip
这是一种压缩档，可以用 pkunzip 来解压，用 pkzip 可以造出这种档。
ftp://ftp.edu.tw/PC/Simtel/msdos/zip/pkz204g.exe
⊙ *.zoo
这是一种压缩档。由 zoo 这个程序来处理。
ftp://ftp.edu.tw/PC/Simtel/msdos/zoo/zoo210.exe

Linux下软件安装方法

我们一起在风雨中前行

Linux下软件安装方法

对于Linux初学者来说，安装一个很小的软件恐怕都是一件很让人头疼的事，因为在Linux下安装软件不像在 Windows中那样简单。在Linux中大多数软件提供的是源代码，而不是现成的可执行文件，这就要求用户根据自己系统的实际情况和自身的需要来配置、编译源程序后，软件才能使用。多数初学者往往不知道该如何进行配置和编译就盲目地运行一些有执行属性的文件或者机械地运行“make”、“make install”之类的命令。结果呢？是软件没装成，自己急出一身汗，后果严重的还会破坏系统的稳定性。下面笔者将安装软件方面的一些规律写出来与大家分享。

目前流行的软件包有两种比较常见的形式，一种是RPM包的形式，另一种是压缩成*.tar.gz的形式。本文将讨论这两种形式的软件包在文本环境和图形环境（X Window）下不同的安装方法。

文本环境下的软件安装
一、安装简便的RPM包
RPM是RedHat Package Manager的缩写，它只能使用在安装了RPM软件的系统中，RedHat Linux和Turbo Linux中已经使用了它。这种结构的包使用起来还是很方便的，只要记住几条简单的命令和参数就可以方便地使用：
#〉rpm [options] filename.rpm

其中常用的options包括：

-i： 安装软件

-e： 卸载软件

-q：查看软件安装的信息和状态

-U： 升级现有软件

例如安装软件时，可以使用如下命令：“rpm -i filename.rpm”，软件安装在什么地方、是怎么安装的都不需要用户操心，RPM可以帮助用户管理。由于RPM使用方便，很多软件都有RPM版本。如果想使用RPM形式的软件就要首先下载一个RPM管理软件。在ftp://ftp.rpm.org/pub/rpm/dist/rpm-version

二、安装需要编译的.tar.gz包

由于RPM包使用的局限性（必须安装RPM），目前更多的软件使用的是源代码形式的.tar.gz包。这种软件的安装通常要经过解压缩、软件配置、软件编译及安装的过程。

解压缩通常有两种命令方法：一种是“gunzip filename-VERSION-OS.tar.gz | tar xvf -”，它实质是两条命令“gunzip
filename-VERSION-OS.tar.gz”和“tar xvf filename-VERSION-OS.tar”；另一种是“tar
xzvf filename-VERSION-OS.tar.gz”。 一般来说在软件解压缩后会生成一个目录filename-VERSION-OS。

软件的配置、编译、安装是最让初学者望而生畏的事了，但笔者认为掌握一些规律还是不难的。一般来说，在解压缩生成的目录中都会有名为Readme、Rnstall或Readme.install之类的文件。这些文件通常会对软件的功能、特性、版权许可、安装以及相关知识加以介绍，并且会提到关于安装的方法和步骤。举例来说：在apache_1.3.6的install文件中说明了如下内容（此处只列出总的条目，具体内容省略）：

Installing the Apache 1.3 HTTP server with APACI

==============================

1.Overview for the impatient（概括说明配置的全过程）

$./configure--prefix=PREFIX

$make

$make install

$PREFIX/bin/apachectl start

2.Requirements（需要的条件）

3.Configuring the source tree(配置的参数说明)

4.Building the package（编译软件的方法）

5.Installing the package（安装软件的方法）

6.Testing the package（软件测试）

理解并能熟练使用这些说明文件后，就可以利用一些规律来安装大多数的软件。对于那些没有说明文件的软件（当然这种情况比较少见），这些规律通常也是适用的。一般来说，与安装软件有直接关系的文件只有两个：configure
、Makefile。

其中，configure文件具有可执行的属性，是用来配置软件的，它的参数比较多，用法也比较灵活。当然，不同的软件参数也不相同，这时候就需要借助它的help参数，运行下面的命令就会让你感到豁然开朗：

#〉 ./configure -help

Usage: configure [options]

Options: [defaults in brackets after descriptions]

General options:
--quiet, --silent do not print messages
--verbose,-v print even more messages
--sha [=DIR] switch to a shadow tree (under DIR) for building
Stand-alone options:
--help,-h print this message
--show-layout print installation path layout (check and debug)

图1 Gnome运行界面
Installation layout options:
--with-layout=[F:]ID use installation path layout ID (from file F)
--target=TARGET install name-associated files using basename TARGET
……

接下来，就可以运行“./configure [options]”来配置该软件。注意，命令行中的“./”非常重要，它告诉系统要运行的命令就在当前目录下（否则系统就会到$path变量指定的路径下去执行命令）。执行命令后可以生成Makefile文件或者修改已有的文件配置。

Makefile文件通常是用来编译和安装软件的。运行make命令时系统会自动根据Makefile文件中的设置对软件进行编译和安装。make命令有时还可以带一些参数，如：all、build、config、install等。具体要带哪个参数可以参看Makefile文件。在Linux中绝大部分文件是文本文件，Makefile就是一个shell程序（Linux中shell程序与DOS中的批处理文件有很多相似之处，当然功能要强得多），很容易读懂，尤其是编译时可带的参数都会明确写出，例如：

```
##========================
## Targets
##========================
# default target
all: build
##------------------------
## Build Target
##------------------------
# build the package
build:
……
# the non-verbose variant for package maintainers
build-quiet:
 @$(MAKE) -f $(TOP)/$(MKF) $(MFLAGS) $(MFWD) QUIET=1
```

```
build
    # build the additional support stuff
    build-support:
    ……
    ##------------------------
    ## Installation Targets
    ## -----------------------
    # the install target for installing the complete Apache

    # package. This is implemented by running subtargets
for the
    # separate parts of the installation process.
    install:
    ……
    # the non-verbose variant for package maintainers
    install-quiet:
     @$(MAKE) -f $(TOP)/$(MKF) $(MFLAGS) $(MFWD) QUIET=1
install
    # create the installation tree
    install-mktree:
    ……
```

上面这段代码是apache_1.3.6的Makefile文件的一部分，从这段程序可以看出all参数表示完全编译（缺省参数）。此外，编译时还可以带build、build-quiet、build-surpport等参数；安装时可以带install、install-quiet、install-surpport等参数。它们的功能分别在“#”表示的注释中进行

说明。需要额外说明的是，有些软件（例如Linux的内核升级程序）不用configure命令来配置软件，而是用make config来完成这项工作，所以，具体使用哪种方法要具体问题具体分析。

图形界面下的软件安装

在图形环境下，同样可以弹出一个仿真终端以文本的方法来安装软件，但那样就太笨了，因为在X Window中有一些简单的方法可以帮您完成软件安装。下面笔者以RedHat
6.0为例做介绍。

一、图形界面下安装.tar.gz包

在X Window下，安装这种形式的包简化程度并不大，只是在解压缩软件时方便一些。用鼠标左键双击要安装的软件包，系统就会自动生成一个目录——filename.tar.gz#utar，在这个目录下就有你要解压缩的内容——filename目录。将该目录拷贝到你要解压缩的目录下，解压缩的工作就这样简单地完成了。不过，剩下的工作还要弹出一个仿真终端以文本的方法来完成。

二、图形界面下安装RPM包

在X Window中你要安装、升级、卸载和查询一个RPM软件包实在是太容易了。以Redhat 6.0为例，它的X Window中有一个Gnome
RPM软件可以完成上面提到的一切工作。

点击“Start”→“System”→“GnomeRPM”，可以运行该软件。它将安装好的RPM包形式的软件按照功能分在Amusements、Application、Development、Document、Extension、Extentions、Networking、System
Environment、UserInterface、X11等几个树形目录中，每个目录中有相应的文件图标和名称。要安装或升级软件，只要点击工具栏的相应按钮就会弹出打开文件的对话框，选中你要安装的RPM文件，单击“OK”按钮，一切大功告成；卸载软件时，需要根据分类找到该软件的图标，点击右键，选Uninstall就可以了。如果你会在Win
95/98中查找文件的话，在Linux中查找已经安装的软件也就不难了。这个软件可以到下面的站点下载：ftp.gnome.org/pub/GNOME/stable/

sources/gnorpm。

几点注意事项

一、安装方法的适用范围

上面提到的软件安装方法并不是在任何版本的Linux上都适用。笔者认为，.tar.gz包的安装方法适用于各种版本的Linux，而RPM包则有一定的局限性。

目前常见的各种Linux发行版本中，如：Redhat 6.0、Turbo Linux 3.0.2、Xteam 3.0等都支持RPM包。如果你想知道你使用的Linux是否支持RPM包，只要运行一下“rpm
--help”命令就知道了。不过，对于不支持RPM包的版本，可以安装一个RPM管理软件。

二、容易出现的问题

在安装软件时，一定要保证你对用到的软件包有访问权限。当然如果你是root就没问题了。但如果你真的是root你就需要注意另一个问题，由于root的权限过高，所以在安装软件时，要防止对系统其它软件造成误操作（在使用rm
等“危险”命令时，尤其要注意）。

另一个容易出现的问题是，在卸载RPM包的软件时要慎重，因为很多软件之间是相互关联的，你卸载的软件很可能是其它软件要用到的，要防止由于卸载了一个软件而影响另一个软件的正常使用。所以笔者建议，初学者对于与系统运行有关的软件尽量不要删除（对于游戏、应用软件一类的软件不必有太多顾虑）。等你成为一名经验丰富的系统管理员时，你就能灵活地处理这些问题了。

Linux与其他操作系统的区别

我们一起在风雨中前行

Linux与其他操作系统的区别

Linux可以与MS-DOS、OS/2、Windows等其他操作系统共存于同一台机器上。它们均为操作系统，具有一些共性，但是互相之间各有特色，有所区别。

目前运行在PC机上的操作系统主要有Microsoft的MS-DOS、Windows、Windows NT、IBM的OS/2等。早期的PC机用户普遍使用MS-DOS，因为这种操作系统对机器的硬件配置要求不高，而随着计算机硬件技术的飞速发展，硬件设备价格越来越低，人们可以相对容易地提高计算机的硬件配置，于是开始使用Windows、Windows
NT等具有图形界面的操作系统。Linux是新近被人们所关注的操作系统，它正在逐渐为PC机的用户所接受。那么，Linux与其他操作系统的主要区别是什么呢？下面从两个方面加以论述。

首先看一下Linux与MS－DOS之间的区别.在同一系统上运行Linux和MS-DOS已很普遍，就发挥处理器功能来说，MS-DOS没有完全实现x86处理器的功能，而Linux完全在处理器保护模式下运行，并且开发了处理器的所有特性。Linux可以直接访问计算机内的所有可用内存，提供完整的Unix接口。而MS-DOS只支持部分Unix的接口。

就使用费用而言，Linux和MS-DOS是两种完全不同的实体。与其他商业操作系统相比，MS-DOS价格比较便宜，而且在PC机用户中有很大的占有率，任何其他PC机操作系统都很难达到MS-DOS的普及程度，因为其他操作系统的费用对大多数PC机用户来说都是一个不小的负担。Linux是免费的，用户可以从internet上或者其他途径获得它的版本，而且可以任意使用，不用考虑费用问题。

就操作系统的功能来说，MS-DOS是单任务的操作系统，一旦用户运行了一个MS-DOS的应用程序，它就独占了系统的资源，用户不可能再同时运行其他应用程序。而Linux是多任务的操作系统，用户可以同时运行多个应用程序。

再看一下Linux与OS/2、Windows、Windows NT之间的区别。
从发展的背景看，Linux与其他操作系统的区别是，Linux是从一个比较成熟的操作系统发展而来的，而其他操作系统，如Windows NT等，都是自成体系，无对应的相依托的操作系统。这一区别使得Linux的用户能大大地从Unix团体贡献中获利。因为Unix是世界上使用最普遍、发展最成熟的操作系统之一，它是七十年代中期发展起来的微机和巨型机的多任务系统，虽然有时接口比较混乱，并缺少相对集中的标准，但还是发展壮大成为了最广泛使用的操作系统之一。无论是Unix的作者还是Unix的用户，都认为只有Unix才是一个真正的操作系统，许多计算机系统（从个人计算机到超级计算机）都存在Unix版本，Unix的用户可以从很多方面得到支持和帮助。因此，Linux做为Unix的一个克隆，同样会得到相应的支持和帮助，直接拥有Unix在用户中建立的牢固的地位。

从使用费用上看，Linux与其他操作系统的区别在于Linux是一种开放、免费的操作系统，而其他操作系统都是封闭的系统，需要有偿使用。这一区别使得我们能够不用花钱就能得到很多Linux的版本以及为其开发的应用软件。当我们访问Internet时，会发现几乎所有可用的自由软件都能够运行在Linux系统上。有来自很多软件商的多种Unix实现，Unix的开发、发展商以开放系统的方式推动其标准化，但却没有一个公司来控制这种设计。因此，任何一个软件商（或开拓者）都能在某种Unix实现中实现这些标准。OS/2和Windows
NT等操作系统是具有版权的产品，其接口和设计均由某一公司控制，而且只有这些公司才有权实现其设计，它们是在封闭的环境下发展的。

2005-1-20 上一页 首页 下一页

Red Hat 7.3汉化

我们一起在风雨中前行

linux完全汉化汉化

由于linux版本众多，在这里我就以我汉化RedHat7.3 的过程为例，我将在不久的以后专门制作一个关于汉化的CHM文档。在

Linux 的字体已经比较漂亮，中文字体用宋体，英文用 Windows 2000 的 Tahoma 字体。为了做到这点，我将宋体和 Tahoma

“合并”为一个新的字体－simsuntahoma，另外，为了使用 qterm 上 BBS 好看，还单独做了宋体。

(1) 安装时将简体中文选上（为了能够浏览繁体的网站，建议将繁体中文也选上(我建议不选，因为我的经验是选上后中文会有乱码----转载者注），

将默认语言设置为简体中文，默认桌面系统为 KDE。

(2) 从 Windows 下拷出 simsun.ttc，tahoma.ttf，tahomabd.ttf

到 /usr/share/

fonts/zh_CN/TrueType/simsuntahoma 目录，将 simsun.ttc 改名为 simsun.ttf 在该

目录下建立 fonts.dir，内容如下:

```
25
simsun.ttf -misc-fixed-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0
simsun.ttf -misc-simsuntahoma-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.2:simsun.ttf -misc-simsuntahoma-medium-o-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.4:simsun.ttf -misc-simsuntahoma-medium-i-normal--0-0-0-0-c-0-gb2312.1980-0
ab=y:simsun.ttf -misc-simsuntahoma-bold-r-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.2:ab=y:simsun.ttf -misc-simsuntahoma-bold-o-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.4:ab=y:simsun.ttf -misc-simsuntahoma-bold-i-normal--0-0-0-0-c-0-gb2312.1980-0
tahoma.ttf -misc-simsuntahoma-medium-r-normal--0-0-0-0-p-0-iso8859-1
ai=0.3:tahoma.ttf -misc-simsuntahoma-medium-i-normal--0-0-0-0-p-0-iso8859-1
tahomabd.ttf -misc-simsuntahoma-bold-r-normal--0-0-0-0-p-0-iso8859-1
ai=0.3:tahomabd.ttf -misc-simsuntahoma-bold-i-normal--0-0-0-0-p-0-iso8859-1
ai=-0.4:simsun.ttf -misc-simsuntahoma-medium-ri-normal--0-0-0-0-c-0-gb2312.1980-0
ai=-0.4:ab=y:simsun.ttf -misc-simsuntahoma-bold-ri-normal--0-0-0-0-c-0-gb2312.1980-0
ai=-0.2:simsun.ttf -misc-simsuntahoma-medium-ro-normal--0-0-0-0-c-0-gb2312.1980-0
```

```
ai=-0.2:ab=y:simsun.ttf -misc-simsuntahoma-bold-ro-normal--0-0-0-0-c-0-gb2312.1980-0
simsun.ttf -misc-simsuntahoma-medium-r-normal--0-0-0-0-c-0-gbk-0
ai=0.2:simsun.ttf -misc-simsuntahoma-medium-o-normal--0-0-0-0-c-0-gbk-0
ai=0.4:simsun.ttf -misc-simsuntahoma-medium-i-normal--0-0-0-0-c-0-gbk-0
ab=y:simsun.ttf -misc-simsuntahoma-bold-r-normal--0-0-0-0-c-0-gbk-0
ai=0.2:ab=y:simsun.ttf -misc-simsuntahoma-bold-o-normal--0-0-0-0-c-0-gbk-0
ai=0.4:ab=y:simsun.ttf -misc-simsuntahoma-bold-i-normal--0-0-0-0-c-0-gbk-0
ai=-0.4:simsun.ttf -misc-simsuntahoma-medium-ri-normal--0-0-0-0-c-0-gbk-0
ai=-0.4:ab=y:simsun.ttf -misc-simsuntahoma-bold-ri-normal--0-0-0-0-c-0-gbk-0
ai=-0.2:simsun.ttf -misc-simsun-mediumtahoma-ro-normal--0-0-0-0-c-0-gbk-0
ai=-0.2:ab=y:simsun.ttf -misc-simsuntahoma-bold-ro-normal--0-0-0-0-c-0-gbk-0
```

然后 cp fonts.dir fonts.scale，或者 cat fonts.dir > fonts.scale

(3) 将 simsun.ttf 复制到 /usr/share/fonts/zh_CN/TrueType/simsun
目录，类似地，
建立 fonts.dir 文件，内容如下:

```
31
simsun.ttf -misc-fixed-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0
simsun.ttf -misc-simsun-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.2:simsun.ttf -misc-simsun-medium-o-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.4:simsun.ttf -misc-simsun-medium-i-normal--0-0-0-0-c-0-gb2312.1980-0
ab=y:simsun.ttf -misc-simsun-bold-r-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.2:ab=y:simsun.ttf -misc-simsun-bold-o-normal--0-0-0-0-c-0-gb2312.1980-0
ai=0.4:ab=y:simsun.ttf -misc-simsun-bold-i-normal--0-0-0-0-c-0-gb2312.1980-0
sw=1:bw=0.5:simsun.ttf -misc-simsun-medium-r-normal--0-0-0-0-c-0-iso8859-1
ab=y:sw=1:bw=0.5:simsun.ttf -misc-simsun-bold-r-normal--0-0-0-0-c-0-iso8859-1
ai=0.2:sw=1:bw=0.5:simsun.ttf -misc-simsun-medium-o-normal--0-0-0-0-c-0-iso8859-1
ai=0.4:sw=1:bw=0.5:simsun.ttf -misc-simsun-medium-i-normal--0-0-0-0-c-0-iso8859-1
ai=0.2:ab=y:sw=1:bw=0.5:simsun.ttf -misc-simsun-bold-o-normal--0-0-0-0-c-0-iso8859-1
ai=0.4:ab=y:sw=1:bw=0.5:simsun.ttf -misc-simsun-bold-i-normal--0-0-0-0-c-0-iso8859-1
ai=-0.2:sw=1:bw=0.5:simsun.ttf -misc-simsun-medium-ro-normal--0-0-0-0-c-0-iso8859-1
ai=-0.4:sw=1:bw=0.5:simsun.ttf -misc-simsun-medium-ri-normal--0-0-0-0-c-0-iso8859-1
ai=-0.4:simsun.ttf -misc-simsun-medium-ri-normal--0-0-0-0-c-0-gb2312.1980-0
ai=-0.4:ab=y:simsun.ttf -misc-simsun-bold-ri-normal--0-0-0-0-c-0-gb2312.1980-0
ai=-0.2:simsun.ttf -misc-simsun-medium-ro-normal--0-0-0-0-c-0-gb2312.1980-0
ai=-0.2:ab=y:simsun.ttf -misc-simsun-bold-ro-normal--0-0-0-0-c-0-gb2312.1980-0
simsun.ttf -misc-simsun-medium-r-normal--0-0-0-0-c-0-gbk-0
ai=0.2:simsun.ttf -misc-simsun-medium-o-normal--0-0-0-0-c-0-gbk-0
ai=0.4:simsun.ttf -misc-simsun-medium-i-normal--0-0-0-0-c-0-gbk-0
ab=y:simsun.ttf -misc-simsun-bold-r-normal--0-0-0-0-c-0-gbk-0
ai=0.2:ab=y:simsun.ttf -misc-simsun-bold-o-normal--0-0-0-0-c-0-gbk-0
ai=0.4:ab=y:simsun.ttf -misc-simsun-bold-i-normal--0-0-0-0-c-0-gbk-0
ai=-0.4:simsun.ttf -misc-simsun-medium-ri-normal--0-0-0-0-c-0-gbk-0
ai=-0.4:ab=y:simsun.ttf -misc-simsun-bold-ri-normal--0-0-0-0-c-0-gbk-0
ai=-0.2:simsun.ttf -misc-simsun-medium-ro-normal--0-0-0-0-c-0-gbk-0
ai=-0.2:ab=y:simsun.ttf -misc-simsun-bold-ro-normal--0-0-0-0-c-0-gbk-0
```

cp fonts.dir fonts.scale 或者 cat fonts.dir > fonts.scale，生成
fonts.scale

(4) 进入 /etc/X11 目录，修改 XF86Config-4
文件，

a. 找到 FontPath 行，在 FontPath "unix/:7100" 一行前面（注意要在这个前面）
加上 FontPath "/usr/share/fonts/zh_CN/TrueType/simsuntahoma",
FontPath "/usr/share/fonts/zh_CN/TrueType/simsun"

b. 找到 modules 段，在这些 Load 行里面加上一行 Load "xtt"，注意，这个也要放在第一行。如果有
Load "type1"， Load "freetye" ，注释掉。

(5) 到这里，系统就多了 simsuntahoma
和 simsun 字体了，而且还有粗体和斜体。
在 KDE 控制中心内，观感-->字体内改所的有字体为 simsuntahoma，字号 12，这样系统 的中文就变成了宋体，而英文则变成了
Tahoma 字体，相当 PP.

(6) 下面谈谈应用程序的汉化问题：

a. 为了让 qterm 能够显示漂亮的宋体，有了前面的基础就好办了。在主目录下.bashrc 文件加入：

export LANG=zh_CN.gb2312

export LC_ALL=zh_CN.gb2312

然后在站点属性里面设置字体为 simsun（注意，这里不要用 simsuntahoma），这样就可以开开心心地上 BBS 了，爽～

b. 像 xmms 这种程序，字体需要 fixed 的，修改/etc/gtk/gtkrc.zh_CN 就可以了，将其中的
fontset 改为
fontset = "-misc-fixed-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0,\
-misc-simsuntahoma-medium-r-normal--0-0-0-0-p-0-iso8859-1"
这样在 xmms 里面就可以看到 pp 的中文菜单了 :P，另外，为了让 xmms 正确显示中文歌名，需要到参数设置里面将字体设置为

-misc-simsun-medium-r-normal--0-0-0-0-c-0-gb2312.1980-0

确定以后，你会发现歌名都正确显示了，而且比较 PP

c. Mozilla 中文支持:

首先选择 Edit--Preferences，弹出对话框，选择 Appearance--Fonts，在
Language Encoding 中选择 Simplified Chinese，然后将 Serif、Sans Serif、
Monospace 都选择misc-simsun-gb2312.1980-0 字体，字号设为 14。

有些站点可以自动识别语言。这可以通过以下方法实现：Navigator--Languages--Add...
,选择 Chinese/China [zh-cn], 点击Move Up，将中文语言放在最上面。

最后点OK完成配置，然后退出Mozilla。（注意：这一点很重要，因为mozilla退出时会改写配置文件。如果没有退出mozilla做以下操作将无效！）

cd /usr/lib/mozilla/defaults/pref，编辑unix.js

找到这样两行

pref("font.min-size.variable.zh-CN", 16);

pref("font.min-size.fixed.zh-CN", 16);

改为

pref("font.min-size.variable.zh-CN", 12);

pref("font.min-size.fixed.zh-CN", 12);

这样可以使网页看上去美观多了。

最后是对输入法的配置

cd ~/.mozilla

在里面找到pref.js文件，在文件最后加上这样一行

pref("xim.input_style", "over-the-spot");

然后运行 Mozilla，就可以显示效果很好的中文网页了！显示效果一点不比 IE 5.0 差。

为了方便中文识别，可以选择View--Character Coding--Auto-Detect--Simplified Chinese 以后 Mozilla 会自动识别中文代码页面。

(7) 到这里，KDE 就相当 PP 了，感谢 SMTH 的 fiasco 和 benben00 的帮助，参考了 http://www.linuxaid.com.cn及 http://www.linuxeden.com 的相关教程。

Redhat 7.2完全安装手册

我们一起在风雨中前行

linux完全安装手册

Linux的版本有很多，这里我们仅以Redhat 7.2为例来介绍Linux的安装。

Red Hat图形安装界面(Graphical mode) 使用光盘或是boot. img启动盘启动之后，您会看到Red Hat启动选单，上面的信息告诉您只要是Red Hat 3. 03以后的版本，都可以直接升级成Red Hat 7. 2版，另外Red Hat提供一些功能键F1—F5，可以观看不同安装模式的on-line HELP，最下面会出现LILO boot：提示符号，您可以在此输入某些关键字选择安装模式或加入设置Kemel参数，例如：内定安装模式就是graphical mode，采用图形操作界面安装Red Hat,直接按“ENTER”即可进入；text mode(boot:text)就是传统的文字选单安装模式：另外有exprt mode(boot：expert,专家模式)等三种. 至于rescue mode(boot:linux rescue)使系统修复模式，此种模式是用来为维护安装好的linux系统，我们初次安装Linux系统自然不能选择这种模式。最后一项DriverDisk(boot:linux dd)则是专为拥有特殊硬件的电脑设计的安装模式，若boot. img制作的磁盘没有办法正确搜索到所有的硬件，则可以改采用此种安装法，因为driver disk含有特殊硬件的驱动程序。但是Driver Disk安装法必须另外制作driver disk磁盘，该磁盘的image file放置在CD1的images / drivers. img，也就是说您必须同时制作boot. img和drivers. img这两张盘，才可以选择linux

dd这种安装模式。一般若没什么特别需要，请采用fraphicalmode图形安装模式，按“ENTER”启动Linux吧!不过在此必须要注意的是，若您的显卡XFree86
X Server不支持，则无法启动图形安装模式，此时Red Hat会自动切换到text mode，不过安装方法是完全一模一样的，请不用担心。还有若您内存大于128MB，而Linux没有完全正确搜索的话，可指定内存大小给Kernel.例如：boot：linux
mem＝128M expert。一切准备就绪后,接着请参考下列步骤，一步步做下去，您会发现安装RedHat是如此简单快速。进入比Red Hat安装画面后，若PC有安装SCSI卡，则会自动驱动这块SCSI卡；若没有，则会直接启动X
Window System进行安装。启动安装程序后，您可利用功能键切换各终端机，查看各种安装信息，例如：Ctrl+Alt+F7就是最主要的安装画面，您可以切换到Console
1看RedHat安装信息。

Console 1:Ctrl+Alt+F1 installation dialog

Console 2:Ctrl+Alt+F2 shell prompt

Console 3:Ctrl+Alt+F3 install log (messages from installation program)

Console 4:Ctrl+Alt+F4 system-related messages(messages from the kernel,etc.)

Console 5:Ctrl+Alt+F5 other messages

Console 7:Ctrl+Alt+F7 X graphical display

Stepl. 选择语系

Red Hat有提供多国语言的安装界面，但是都是英语系或欧洲语系国家，并没有提供中文的安装界面，所以请选择English。如图是选择Red
Hat语系画面，画面左边是网上辅助说明，若不想看可以单击Hide Help将此Online Help隐藏；右半部则是语系选单，单击确定之后，单击Next继续进行安装。而这个GUI按装界面除了可以用鼠标操作外，也可使用键盘的Tab切换选项，用“Enter键确定。

Step 2. 选择键盘

键盘一般都兼容于U. S. Englishl01键的键盘，所以就选内定值PC即可，如图。若是此时选择了错误的键盘模式，可以立即单击设，或者在完成Red
Hat Unux安装之后，以rodt身份登录，执行／u或是setup重新设置。

Step 3. 选择鼠标

鼠标一般都是两键／三键的PS/2、Bus Mouse或是Serial鼠标，如图2—5选择鼠标种类与连接端口。若您使用PS／2或Bus
Mouse，则不必选择连接端口；使用Serial鼠标则必须设置正确的连接端口，例如：CoM1 port在Linux系统看来就是ttyS0这个装置名称，COM2
port就是ttyS1，请选择正确连接端口才能运行正常。最下面有个Emulate 3 Buttons可以让两键的鼠标左右键一起按时，模拟三键鼠标的中键。同样，若设置鼠标错误；可在安装之后，执行/usr/sbin/mouseconfig重新设置。假若您是左撇子或惯用左手鼠标，可以在安装好Red
Hat之后，执行gpm-b 321，设置鼠标为左手鼠标。

Step4.Red Hat安装欢迎画面

看到可爱的小红帽欢迎画面了吧!这里简介安装方法详细记录在Red Hat Linux Installation Guide, 有HTML和PostScript两种文件格式，
并可从http:／／www. redhat. com网站取得，其实一般Red Hat光盘的doc/目录都有安装手册。另外看到画面下面有System

Installer的字眼吧!Linux又这么漂亮的安装画面都是归功于RedHat公司编写loader安装程序的功劳(该程序在/sbin/loader)。红的帽子Linux(PC机用的免费UNIX操作系统)。请在左产品上的信息读完帮助正文。在屏幕的底部的左角有一个隐藏的帮助的按钮。帮助屏幕在没有特别指定的情况下是开着。如果你不想看帮助信息，可以单击隐藏帮助。继续单击下一步。

Step 5.选择Red Hat安装模式（请特别小心，以免数据全毁）

Red Hat的安装模式分四种：Workstation Server laptop Custom.这里你必须千万小心，因为选择Workstation
或Server的话， Red Hat会自动的帮您分割并格式化硬盘，若数据全毁的话，后悔就来不及了。但是若您预备使用完整的一个硬盘来安装 Red
Hat，并不怕任何格式化的话，则安装和设置会异常顺利。

下面是 Red Hat将安装模式分乘此三者的目的：
Workstation
选择GNOME(GNU Network Object Model Environment)Workstation或是KDE(KDesktop

Envirnment)Workstation的用户，打算将安装好的Linux当workstation使用，X 图形桌面集成环境有GNOME和KDE两种可供选择，但因为该PC只是相当单纯的工作站，所以不会安装任何的网络服务器(例如：web、FTP、News
server等等)，并且Workstation模式会把硬盘上的所有Linux partition全部移除，并重新划分成三个partition：64MB
SWAP partition、/boot(16MB)与/(所有剩余硬盘空间，需要大于900MB。请注意：使用Workstation模式硬盘空间建议至少有1GB)。而后续的安装步骤：硬盘分割、格式化硬盘、选择安装软件套件、设置LILO多重启动系统，全由Red
Hat安装程序包办。不过您也不要太担心，不用怕误选Workstation之后，马上就格式化硬盘了，因为后续画面是Automatic Partitioning，有Remove
data与Manually Partition with DiskDruid/fdisk格式化硬盘的选项，如图所示。只要不选择Remove data, Red Hat就不会自动帮您分割硬盘，但是确定Linux partition硬盘数据不要的情况下，选择Remove data安装会更顺利。另外可让您更放心的一点是，选择Workstation安装模式，绝对不会删除任何的DOS/Windows
9x/windows2000 partition，因此若您的系统已经预先安装Windows98/2000，您只要选择Workstation自动安装模

式，在Linux安装完成之后，还会自动帮您设置LILO成为Windows/Linux多重启动模式，让您可以选择使用Windows或是Linux启动，所以强烈建议初次安装者，选择此种模式安装Linux。

Server
选择Server安装模式的话，请确定您预计安装Red Hat Linux的此个硬盘数据是完全不要的，因为此模式会将硬盘上的所有Partition全部移除!不会橡Workstation模式那么客气只移除Linux partition了，无论是DOS、Windows 9x/NT partition、OS/2全部都删除。然后将硬盘划分为64MB SWAP partition/256MB、/usr至少512MB、/home至少512MB、/var 256MB五个partition。一般情况下，若您硬盘很大，多余的硬盘空间都由/usr与/home两个partition平分，而且您要注意的是，至少拥有1.7GB的硬盘空间才可以采用Server模式安装，而除了硬盘分割方式不同外，Workstation与Server的后续安装步骤都是一样的。而另外一点Workstation与Server安装模式不同的地方为，Server模式不会安装任何的图形界面程序，
目的是为了让各类网络服务器发挥最大的性能，而不要将系统资源消耗在无谓的 X Window System上。

至于server模式划分这么多Partition将数据个别分开放置的目的是：除了可以有效提升server的网络服务器运行性能外，还可以降低数据毁损风险与降低系统重建的复杂度。若读者您不知道这些partition名称为什么是／usr、／home之类的目录名称的话，请阅读第3章与第4章，或暂时先不要了解也没关系。

Custom

Custom安装模式就是凡事自己来，分割硬盘、配置各partition空间大小、格式化硬盘、选择软件套件、多重启动程序LILO的设置全由自己设置。接下来的安装模式教学，是采用Custom选项安装
Red Hat，因为这样才能完整讲解到Red Hat的所有安装设置，让您成为Red Hat Linux的主人。另外若您是第一次安装Red Hat，还是建议您可以先使用workstation自动安装模式，待成功安装过一次后，再自己尝试使用Custom安装模式，如此在学习Linux安装上，可能会较为顺畅。

Upgrade

若您先前已经安装过Red Hat 3．03至6．2的旧版本，都可以选用这个选项升级，Red Hat安装程序会完全保留系统的旧有配置文件，并把文件名附加上.rpmsave的副文件名，例如：／etc／profile就被改成／etc／profile．rpmsave，并且完整的Red
Hat升级信息会被记录在／tmp／upgrade．log以供您参考。

Step 6．分割硬盘

选择Custom之后，可以马上看到所有硬盘的Partition情况，如图所示。所谓的mount
Point就是Linux会将这个partition,mount在您指定的系统目录，要存取这个partition其实就是对这个目录做存取，例如/dev/hdal挂接在／win98，表示安装好Linux之后，IDE
I Master硬盘的第一个partition，会mount在／win98这个目录。这点观念与DOS/Windows的C盘、D盘观念是完全不同的，另一个重要观念是Linux将所有电脑硬件都视为一个装置文体,如hda代表第一条IDE排线上的Master硬盘。

自动分割机械分割

分割硬盘非常简单，只要单击Add就可以新增Partition，选Edit可以编辑己存在Partition的Mount point，Delete当然就是删除Partition了。

Reset则可以让您划分错误时，回复到原始状态，因为真正的Partition Table要等到软件安装步骤后才会真正储存起来。最后有一个选项Make
RAID Device因为是要做partition或磁盘的备份，所以必须有两个硬盘以上才可以制作，除非是重要的Server架设，否则读者一般安装不需要使用Software
RAID功能。图2-10是选择Add partition之后的对话框，Mount Point可以是斜线(/)加任意名称，其实斜线(/)是root
partition的意思，一个Linux系统有／与SWAP partition两个Partition就可以运行了。但建议您采用Server安装模式的分割方法：／boot(16MB)、/(256MB)、／usr(建议2GB，视安装软件数目多少决定)、/home(依用户人数多少决定，0～无穷大)、／var(256MB)、SWAP
partition建议128MB以上，最大不要超过256MB，否则浪费。总共必须分割六个partition，决定好mount之后，再在Size(Megs)输入该Partition的空间大小，以MegaBytes为单位，下面Grow
fill todisk是用掉所有硬盘空间的意思，通常在您分割到最后一个Partition时，就可以随便输入Size(Megs)的值，然后选择Grow
fill to disk，便会自动用掉剩余的硬盘空间。最后有一个重要的选项
是Partition Type，所有的Partition都是Linux Native的Ext2文件系统，只有SWAP那个Partition必须

选择为Linux
SWAP，而且SWAP是不需要mount point的，因为它是虚拟内存。

全部Partition规划好之后，下一个画面会让您选择哪几个Partition需要格式化，请将刚刚前面所规划的Partition全部格式化。

Step 7. 安装LILO(Linux Loader)

LILO除了用来载入Linux Kernel启动系统之外，还有一个重要功能就是可以当作多重启动程序，DOS、Windows9x／2000／NT、OS2、FreeBSD、Solaris等等常见的操作系统都可以使用LILo启动。

此选单最上面有个Creator boot disk选项指定要制作启动盘，还有一个Do notinstall
LILO，请不要选择。因为若您不安装LILo就无法直接从硬盘开启Linux，所以建议一定要安装。接着的选项是指定LILO安装位置

MBR:MBR位于Master硬盘、第0磁柱(cylinder)、第0磁头内的第一个扇区(sector)，电脑启动电源后，BIOS会读取这区域去启动操作系统，所以将LILO安装这里可以作为多重启动系统，例如选择用Linux、Windows
9x／NT、DOS或其他OS启动。boot：所谓的boot partition(superblock)，就是硬盘每个Partition的第一个sector。把LILO装在Linux的／(root)boot
partition，可以使用其他的多重启动系统如： System Commander、OSBS、Spfdisk或其他partition上的LILO来启动。而画面的最下面是显示可以启动的Partition有哪些，您可以自己指定启动系统选项(boot
label)以及预设的启动系统(排在第一个的OS)。

Step 8引导装入程序口令构造

Step 9.TCP/IP

Red Hat会自动检测您的网卡并进入TCP/IP网络设置。您必须输入下列网络设置值(以笔者的机器当范例)：最后面的第二台与第三台name
server两项，若您不知道或没有，则可以不用填。若您根本没有固定IP而是使用Intranet，例如：modem用户，则IP Address请使用Intermet保留区段192. 168. xxx. xxx，hostname可以随便您填，Gateway与DNS则填写所使用ISP提供的服务器。

Step 10.Internet 防火墙构造

Red Hat Linux(PC机用的免费UNIX操作系统)向改良的系统安全提供Internet 防火墙保护。在你的计算机和网络之间，网络上的用户能在你的计算机上接近Internet
防火墙。

Step 11.选择语言支持

Red Hat Linux能在你的系统上安装语言,你可以选择全部安装或只选择china，如果你选择安装其他的语言，你能在安装后面改变你的语言设置。

Step 12.设置系统时间

首先您会看到一张漂亮的世界地图，只要点选上面的地名就可以设置当地的系统时间，如图。若是嫌世界地图太小了，可以点选world下拉列表，然后选择Asia,再将鼠标移到中华人民共和国地图上，就可以正确地设置北京时间。不过下面有一个选项“System
clock uses UTC”这一项，千万不要选择，不然您的系统时钟就会被设置成格林威治标准时间，这跟我们目前在北京的时间不符合。若是不小心将系统时间设置错了，待完成安装之后，执

行/usr/sbin/timeconfig即可重新设置。

Step13．新增Linux系统用户

直接在安装过程中新增系统用户是Red Hat的独特创举，不像以往的任何UNIX系统，安装后系统唯一用户是Super User(root)，没有任何一般用户，现在您直接可以在安装系统中设置自己的账号，第一次启动系统不一定非用root登录不可，大幅增加系统安全性。在图账号设置对话框中，最上面的Root Password与Confirm要求您输入一次root密码并再确认一次，所以总共需输入两次root的密码。

Accountname是用户的登录账号，同样地用户密码必须连输两次，最后的Full name可以是任何字符串，通常是输入用户的姓名或数据。

对于初次使用Linux或UNIX系统的读者，笔者在此有个建议，root密码的设置千万不要太简单，最忌讳的就是用自己的英文名字、字典常用字、生日、电话号码等当密码，否则很容易被破解。一个好的密码应该同时包含大小写英文字母(不要是自己的名字或字典的字)、数字以及特殊符号，且不要太长而超过8个字符，以方便记忆。例如Yu＆％Ko4i就不是一个好拆解的密码，同时包含有大小写、数字与特殊符号，虽然没有秩序可言，但短短8个字符常输就会记得，例如：ABC123、youngman、123456、root就是非常不好的密码。

Step14．密码加密与网络安全认证

以往所有用户的数据与密码(加密后)都放在／etc／passwd，但是人人可以读取，所以即便密码稍微加密过，安全性还是非常堪忧，为了增加系统安全性，现在大部分系统都采用Shadow
Password，将假密码放在／etc／passwd，真实的用户密码会放在／etc／shadow这个文件，．且只有唯一的super User－root可以观看或编辑这个文件。但是有人觉得这样还是不够安全，所以再采用MD5密码加密技术，将任何长度的密码，编码成256字符长度后，再存放到／etc／shadow中。所以为了增加系统安全性，请务必将Enable
MD5 passwords与Enable shadow passwords两个选项选取起来(内定值)。另外还有一个Enable NIS,这就和密码加密不太有关系，NIS是Network
Information System的缩写，而NISServer也就是常听的SUN Microsystems公司发明的YP Server，该网络Server主要功能是可以将分散的工作站集中到一台主要的NIs
server管理，例如有cc1~cc10共10台机器，若要让用户使用这10台机器，就必须在每台机器新增这些用户。但是使用了NIS后，假设cc1是NIS
server，其他cc2~cc10都是NIS client，则只需在cc1此台server建立账号或做其他用户设置，则这些用户的基本数据，如username,user
grouP，homedirectory，disk quota在这10台都一样，因为所有的数据都记录在NIS server，其他NISclient需要时会自动向server索取，大幅增加账号管理的方便性。但是若您是单机，则不须要启动NIS，所以忽略此选项。另外下面还有两个选项Enable
LDAP与Kerberos。LDAP(Lightweight Directory Access Protocol)是一种数据组织汇整系统，可以将您所管理的整个网络系统上的所有用户数据汇整在一起，例如：在一个公司行号内，有20个人在做同一个名为x的project，每一个人做不同的事情，但是information必须互相共享，若Mail
Server有通过LDAP机制，则可设置任何一个人收到关于x project的信件，必须自动转信给其他20个人，而这种具备组织结构转信的功能，就是LDAP提供的功能之一。另外LDAP也可以汇整所有用户的姓名、住址、电话、E-mail
address等等，使所有网络用户数据统一管理。至于Kerberos是一种网络加密认证系统，它可以将Kerberos client/server之间传送的密码加密，而不至于被监听或截取密码。在此为了简化系统安装步骤，而且一般用户也不需要此两项功能，所以LDAP与Kerberos的设置就不加以介绍，若读者有兴趣，可以参考LDAP
Linux HOW—To以及Kerberos 5 Installation Guide，此两份文件可在RedHat安装成功后，在／

usr/doc/HOW-TO以及/usr/share/doc目录中找到。

Step15. 选择软件套件

Red Hat将其所提供的所有软件分类规划，如图所示。例如您想使用打印机，自然必须选取Printer Support这个套件；想使用X Window System与GNOME，就将两者都选择起来。不过事实上，Red Hat都有预设的内定值，请您不要把内定值的套件移除掉，只要再选取您想要安装的套件即可。例如：KDE不是内定选项，您想要将来的系统可以同时有GNOME与KDE桌面集成环境可供切换，就将KDE也安装起来。若想完整安装Red Hat，只要选择Everything即可，但是完整安装大约需要3.5GB硬盘空间。

除了选择大项目的分类套件以外，若您想知道这个大项目里面预设安装了哪些软件可以点选Select individual packages，再单击Next就会看到详细的软件列表，如图所示。您可以再自行选择想安装的软件，若是不知道该软件的功用，下面都有网上说明仔细看一下应该可以很容易了解该软件的功用。但是您自己选择软件套件时，可能会发生的就是软件相依性问题，且通常是欠缺安装某些函数库的问题，例如：图所显示的就是选择安装kdetoys这个软件，但是这个软件执行时需要KDE support软件、KDE Library与Qt Library才能正确执行，所以Red Hat的RPM套件管理程序会自动检测出软件间相依性，并要求您必须安装kdesupport、kdelib、qtlx才能安装kdetoys。还有一个您关心的问题可能是，假若有某个套件在安装时没有选到，该怎么办?别担心，待安装好Red Hat之后，执行rpm或是X Windows下的gnorpm，将目录切换到 Red Hat/RPMS就可以再安装这些软件。

Step16.GNOME简短介绍

GNOME是强大的制图法-的开动的桌面环境。GNOME包括展示板（开始应用，展览状态），桌面（数据和应用能被放），窗教练复数的（控制你的桌面的外观和感觉）和桌面工具和应用的标准的一套在内。

http://www.8bb8.com/photo/setup72/setup20.gif

（以下由于害怕文档过大没有在加图片，只要在安装的过程小心或者以默认一般是没问题的）

Step17.KDE的简短介绍

KDE是也是个强大的桌面系统，很精美！窗口化的操作方式，操作起来非常容易！

Step18.软件包依靠

为了正确地运行Unresolved Dependencies Many软件包依靠必须被安装在你的系统上的其他的软件包。例如，用图表示的红的帽子系统管理工具的很多需要 python and pythonlib packages。你安装，或者除掉软件包每次弄清你的系统有全部它为了是充分地功能，Red Hat Linux需要依赖检查这些包。

Step19.X设置工具

接着安装程序会进入x Windows System的设置，首先是屏幕规格的选择，如图所示。您可依照自己使用的屏幕品牌，从选项中选出适当的屏幕。若是无法找到您所使用的屏幕，可以选择Generic Monitor自己指定屏幕，并在下方的水平频率以及垂直频率中输入符合屏幕规格的显示频率范围。选定屏幕之后，接着选择您使用的显卡，如图所示。

Step20.准备安装

注意：先前分割硬盘的Partition table将在这个画面单击Next之后真正写入硬盘，所以若安装至此您想反悔还来得及。直接重新启动，即可终止安装Red
Hat Linux，硬盘的原数据依然还在。

Step21.进行软件安装

到达这个步骤，安装程序将开始将您先前选择的软件套件复制到硬盘，依照您选择的软件多少，等待时间从10min~30min不等，静心等候吧!如图所示，Total栏中显示总共安装441个软件，约需1018MB的硬盘空间，约要11.49分钟完成安装。而这里要请您特别注意的是，如果您选择Workstation或者是Server安装模式，事实上只需要Red
Hat 7. 2的第一张光盘即可安装完成。只有在您选择Evrything，要求安装全部的套件时，才会使用到第二张光盘。而另一种使用到第二张光盘的情况为，当您选择升级模式(upgrade)时，通常也会要求您放入第二张光盘，因为Red
Hat将一些套件改放到第二张光盘了。所以若您安装成功时，发现系统没有要求放入第二张光盘，请不要惊讶，那是绝对正确的。

Step22. 制作启动盘

先前在设置LILO的地方虽然有选择要制作启动盘，但真正的操作是等所有软件安装完成之后才会进行。现在将3.5寸1.44MB磁盘插入，单击Next就开始制作启动盘.若是突然不想要启动盘了，也可以选择Skip
boot disk creation跳过去，将来安成安装之后，也可以执行mkbootdisk制作启动盘。

Step23.X设置工具

同样地，从选项中挑选出您所使用的显卡厂牌与型号，若是无法从中找到您所使用的显卡，请选择Unlisted Card，并选择该块显卡的内存容量。而该画面下面的几个选项：单击Test
this Configuration这个Buttom按钮之后，会测试您选定的屏幕与显卡是否正确，若正确则可显示x画面，不正确则可以单击Back回去重设.

Customize X Configuration可让用户指定X Window System的显示分辨率，若不指定则一般内定采用16位高彩@1280x1024当设置值，如果有24位的或更高的，在这个画面上你也可以看到！您可以选择该选项然后单击Next，就会进入分辨率设置画面。Use
Graphical
Login是指定将来系统安装好之后，登录系统的画面，采用X图形界面，若X环境设置没有问题，请将Use Graphical Login选起来，这样启动以后，看到的就是漂亮的X视窗Red
Hat登录画面。最后的skip X Configuration是忽略X Window system设置，右边的GNOME图标代表将来X Window
system内定采用的视窗集成环境为GNOME，若您有加装KDE套件，也可以选择KDE当内定的视窗集成环境。一切选定后，您可以再次执行Test
this configuration测试将来系统执行X Window system，若觉得没问题，就进行下一安装步骤。若是不满意设置值，也可以在安装好Red
Hat后，在Terminal中执行Xconfigurator,或是在X Window System中执行XF86Setup重新设置。若是不幸XFree86不支持您使用的显卡，您仅能使用Text
mode安装，则请您在安装完成之后，再尝试执行Xconfigurator设置看看，或是参考第X Window
system，里面将会教您如何使用Xf86config设置XWindows或是升级XFree86以支持您的显卡。

Step24.大功告成

最后一个画面告诉您已经完成Red Hat安装，将重新启动系统，请将软盘与光盘拿出来重启动后，您应该会看到漂亮LILO开启画面。

2005-1-20 上一页 首页 下一页

VI的使用

我们一起在风雨中前行

VI的使用

相信用过Linux的朋友一定不会对vi陌生吧，它是Linux环境中使用最为普遍的全屏幕编文本辑器。但由于我们一般用户最初接触的都是微软的产品，它和DOS下的EDIT和Windows下的记事本的使用方法不尽相似，所以另初学Linux朋友觉得vi很难用，甚至无从下手，下面我就向大家介绍vi的使用方法。

首先需要说明的是，vi分为两种状态，即命令状态和编辑状态，在命令状态下，所键入的字符系统均作命令来处理，如:q代表退出，而编辑状态则是用来输入文字资料的。当你进入vi时，会首先进入命令状态。现在对vi也有了基本的认识，如果你有兴趣请接着往向看。

要进入vi，直接在系统提示符下键入vi <文件名>，当你键入的文件名是已有文件时，则系统自动打开此文件，否则将建立一个新文件。这时你将会看到屏幕左边会出现波浪线~，这就代表该行是空的，没有任何文字，这时系统正在命令状态，怎样切换到编辑状态输入文字呢？按键盘上的Insert键即可，这时我们就可以像使用其它的编辑器一样进行文字的编辑了，功能键也和其它编辑器差不多，下面就是一些功能键的说明，是不是很简单？

说明	功能键
移动光标到所在行的最前面	Home
移动光标到所在行的最后面	End
向下翻一页	Page Down
向上翻一页	Page Up
删除光标所在位置字符	Delete
删除光标所在位置前面的字符	Backspace
移动光标	←↑↓→

另外，vi还支持粘贴与复制，不过用键盘来做实在很麻烦，现在哪台电脑没有鼠标呢？和Windows中

一样，从你要复制的开始位置拖动鼠标到结束位置，这块区域就反白了，再将光标移到你要粘贴的位置，按鼠标中键即可完成粘贴（如果你的鼠标是两个键的，同时按左右键即可模拟三键鼠标的中键），这比Windos里方便吧。也许有的朋友会问，如何进行查找呢？下面我们按键盘上的Esc键切换到命令状态，输入“?<你要查找的字符串>”（不带引号）就执行向下查找操作，而“/<你要查找的字符串>”表示向上查找，键盘上的n表示重复一次，而N表示反方向重复一次。

说了半天，可能你的文件已经编辑完成了，但如何存盘呢？现在我们还是保持在命令状态，按:w按后回车即完成了存盘工作，而退出vi返回到Linux的命令是:q，这两个命令也可以组合使用，如:wq代表存盘退出。

好了，说了半天，可能你的头已经大了，上面讲到的那些只是vi中最常用的功能，至于其它的功能你可以在使用中慢慢体会。顺便说一句，不要强记那些命令，它们看起来多而毫无规律，多用几次自然就熟练了，现在我觉得比Windows中的记事本方便多了

2005-1-20 11:29:54 上一页 首页 下一页

2005-1-20 上一页 首页 下一页

从初学者到编程高手，几种必学的编程语言

我们一起在风雨中前行

从初学者到编程高手，几种必学的编程语言

我自己在学习编程的过途中有的几个疑问，1）什么编程语言我需要学。 2）学多少种才算可以。最后通过我自己学习感受和对其他编程高手（主要是新闻组）的请教，我认为找到了满意我的答案。抛砖引玉，做个参考吧。

对于初学者，我建议从 python 开始学。因为它在语言设计上很整洁，帮助文件也很全面。而且也是 object-oriented
(O-O),尽管我把它作为一个初学者开始的语言，并不代表它的用处和功能不强大。实际上它是个功能强大，可以用来做大的编程工程的语言。
我知道国内对 python 可能还不了解，因为它相对来说也比较新(1990)，到现在有12年的历史了，但是它在国外已经被广泛接受和认同，快速发展。不过我知道邮电出版社最近刚出了一本介绍
python 编程的书，大家可以查询。python 的主页 http://www.python.org
Java 也是个很好的语言来开始学编程。不过它比 python 复杂。国外的大学现在都采用 java 作为电脑专业的凯时语言。总之我觉得除了

python 之外，它也可以作为一个极好的初学者语言。

不过，会了这两种语言，并不代表你就是一个编程员了。学会编程的关键就是 -- “ 编程思想 “ 。 学会怎样用一种很“
广泛的思维“ 去考虑“ 编程问题 “，也就是这种思维是可以放之于任何一种编程语言都可以解决问题的，而不是局限于单纯的一种语言。这样当你需要学习一种新的语言时，你只要学习他的语法等等就可以了，至于怎样用这种语言解决编程问题不是早就在你的脑子里了吗。所以，这也就是为什么很多程序员可以拿到一种语言，一两个星期就可以使用的很熟练了。这，也同样要求你去学几种完全不同的语言。

C 语言是非常有效率的，很多时候你都需要考虑内存的管理等底层的东西。很可惜这些都需要你去花很多的时间去做，问每一个
C/C++ 的程序员，用 C/C++ 编程是不是一个很头疼的事。好消息是，现在的机器都已经很高级，运算的很快，在很多时候我们是不用考虑用 C/C++
去做，忽略它所带来的好处，当然除非这个工程真的很需要处理底层的东西（象 内核的编辑，3-D 图形等）。很多时候 python, java 等都可以实现。另外在现在很多工程所要求的时间都不是足够充裕来用 C/C++ 去做，所以在商业上时间就是金钱，这个代价谁也付不起。

我这里并没有说 C/C++ 不好，你不应该学。相反我觉得我们应该学，但不是作为一个初学者的语言来学。在
C 和 C++ 之间，我推荐先学 C 。.因为它相对来说较小，简单。作为一种底层编程语言，C,你可以学到很直接的内存管理，指针,等。另外，很多语言都是用
C 来设计的， 比如 perl , java, python

其它语言值得一学的有 perl , Lisp(当然你也很一说 effile, proflog)，提到 perl
是因为它的实用性，广泛的应用在网络和系统管理，perl 提倡 "There is always more than one way to do it" 这主要是从 perl 的语法以及它的独特设计上来获得,记得用几行code 你就可以写出很有效率的程序？这也就是为什么我很喜欢perl
了。但是perl 的语法着实让很多人头疼 而且对于用 perl 写的大型程序，维护性不是很好，python 就没有这方面的问题，这也是为什么很多
perl 的用户转向了 python 。很多人用 perl 的目的和用 python 一样，避免在不要求 C 的效率时 使用 C。
Lisp 据说是很有启迪性的语言，会让你成为一个更好的程序员.(我还没有到达那个程度 ：-)

我个人推荐 (python, java, C ,perl, 和 Lisp)，因为他们使用不同的方法来编程，每一种都会在某种程度上交给你些很有价值的东西。

谈到学习编程，单纯的上课和看书只会让你学习语言的特性等，只有学会用编程的思想来思考才是你终生受用的。这些只能依靠
“ 练习 “ 具体就是 1) 读程序 2) 写程序 看似简单，但坚持下来你才会真正体会到，这里没有其他可以投机取巧的方法学习编程了。

至于学多少种，是越多越好吧。但并不是指每个都精通，我想大家也没那么多时间和精力。但在你有了坚实的编程经验和基础，了解其它语言的设计方法、构造会开阔你的眼界，没准什么时候你也可以编出自己的语言(咱中国人编的!!)
:-)

　　要提醒大家的是，如果每种语言总是略学皮毛，然后又急匆匆的跳到另外一种语言，那你就会一事无成，永远彷徨在初学者的门槛边。我想这句话不只是对程序员，对我们来说也是生活中的一个道理。

什么是Linux

我们一起在风雨中前行

什么是Linux

　　简单的说,Linux是Unix克隆(Unix clone)或Unix风格(Unix alike)的操作系统(OS),在原代码级上兼容绝大部分Unix标准(指的是IEEE
POSIX,System V,BSD),是一个支持多用户, 多进程,多线程,实时性较好的功能 强大而稳定的操作系统.
它可以运行在x86 PC,Sun
Sparc,Digital Alpha,680x0,PowerPC, MIPS等平台上,可 以说Linux是目前运行硬件平台最多的操作系统.
Linux最大的特点在于它是GNU(Gnu's Not Unix----有点分形与混沌的意味----无限自包含,简单的说GNU是一种自由软件体系)的一员,遵循公共版权许可证(GPL),秉承"自由的思想,开放的源码"的原则,成千上万的专家/爱好者通过Internet在不断地完善并维护它,可以说Linux是计算机爱好者自己的操作系统.
　　追述Linux的历史直到1990年,Linus Torvalds还是芬兰赫尔辛基大学的一名学生,最初是用汇编语言写了一个在80386保护模式下处理多任务切换的程序,后来从Minix(Andy
Tanenbaum教授所写的很小的Unix操作系统,主要用于操作系统教学)得到灵感,进一步产生了自认为狂妄的想法----写一个比Minix更好的Minix,于是开始写了一些硬件的设备驱动程序,一个小的文件系统,......,这样0.0.1版本的Linux就出来了,但是它只具有操作系统内核的勉强的雏形,甚至不能运行,你必须在有Minix的机器上编译以后才能玩.这时候Linus已经完全着迷而不想停止,决定踢开Minix,于是在1991年10月5号发布Linux
0.0.2版本,在这个版本中已经可以运行bash(the GNU Bourne Again Shell----一种用户与操作系统内核通讯的软件)和gcc(GNU
C 编译器).从一开始,Linus就决定自由扩散Linux,包括原代码,他在comp.os.minix新闻讨论组里发布Linux 0.0.2时写到:

　　"Do you pine for nice days of Minix-1.1, when mem were men and
wrote their own device drivers? Are you without a nice project and just
dying to cut your teeth on a OS you can try to modify for your needs?

Are you finding it frustrsting when everything works on Minix? No more
all-nighters to get a nifty program working? Then this post might be just
for you. "As I mentioned a month ago, I'm working on a free version
of a Minix-lookalike for AT-386 computers. It has finally reached the
stage where it's even usable(though may not be depending on what you want),and
I am willing to put out the sources for wider distribution. It is just
version 0.0.2 ... but I've successfully run bash,gcc,gnu-make,gnu-sed,compress,etc.under
it."

随即Linux引起黑客们(hacker)的注意,通过计算机网络加入了Linux的内核开发,Linux倾向于成为一个黑客的系统----直到今天,在Linux社区里内核的开发被认为是真正的编程.由于一批高水平黑客的加入,使Linux发展迅猛,到1993年底94年初,Linux
1.0终于诞生了! Linux 1.0已经是一个功能完备的操作系统,而且内核写得紧凑高效,可以充分发挥硬件的性能,在4M内存的80386机器上也表现得非常好,至今人们还在津津乐道于此,不过自从2.1.xx系列的内核Linux开始走高端的路子----硬件的发展太快了,但是Linux不会失去它的本色.Linux具有良好的兼容性和可移植性,大约在1.3版本之后,开始向其他硬件平台上移植,包括号称最快的CPU---Digital
Alpha(至少目前主频是最高的).所以不要总把Linux与低档硬件平台联系到一块,Linux发展到今天,这是一个误区,它只是将硬件的性能充分发挥出来而已,Linux必将从低端应用横扫到高端应用!

在Linux的发展历程上还有一件重要的事:Linux加入GNU并遵循公共版权许可证(GPL).此举大大加强了GNU和Linux,几乎所有应用的GNU库/软件都移植到Linux,完善并提高了Linux的实用性,而GNU有了一个根基,我现在也搞不清楚到底是GNU
Linux呢还是基于Linux的GNU. :-) 更重要的是遵循公共版权许可证,在继承自由软件的精神的前提下,不再排斥对自由软件的商业行为(如把自由软件打包以光盘形式出售),不排斥商家对自由软件进一步开发,不排斥在Linux上开发商业软件.从此Linux又开始了一次飞跃,出现了很多的Linux发行版,如
Slackware,Redhat,Suse,TurboLinux,OpenLinux等十多种,而且还在增加,注意你不能说"Redhat
Linux""Suse Linux""Slackware Linux"等等,Linux主要指操作系统内核,对所有发行版内核原代码都是一样的(但集成的内核版本可能因发行时间不同而有所不同).还有一些公司在Linux上开发商业软件或把其他Unix平台的软件移植到Linux上来,如今很多IT业界的大腕如
IBM,Intel,Oracle,Infomix,Sysbase,Corel,Netscape,CA,Novell等都宣布支持Linux!
商家的加盟弥补了纯自由软件的不足和发展障碍,Linux迅速普及到广大计算机爱好者,并且进入商业应用,正是打破某些公司垄断文化圈的希望所在!!

Linux是爱好者们通过Internet协同开发出来的,当然它的网络功能十分强大,比如你可以通过ftp,nfs等来安装Linux,用它来做网关等等.随着Linux的发展衍生出来的应用恐怕出乎Linus本人最初的预料,如有人用它来做路由器,有人来做嵌入式系统,有人来做实时性系统......常有新手问Linux能做什么,其实它不象那些中看不中用的操作系统,不在于你用它能干什么,而在于你想干什么!

Linux的兴起还给人们很多启迪与思考,如集市式软件开发的讨论,又如自由软件的精神......

****

PS:Intel平台上有哪些Linux发布(distribution)

主要有Red Hat、Slackware、Debian、SuSE、OpenLinux、TurboLinux。

（1）Red Hat http://www.redhat.com

以容易安装著称，初学者安装这个版本，遇到挫折的机会几乎是零，如果您对安装Win9x已驾轻就熟的话，Red Hat Linux的安装一定难不倒您。

Red Hat另一个优点是它的RPM（Red Hat Package Manager）；以往在安装软件时，最让使用者伤脑

筋的是：软件在解开压缩前先要新建一个目录，然后将软件搬进去解压，解压后，有些部份可能需要搬到另一个目录中去，当要搬移的项目多时，做这些工作就是件苦差事了。而RPM「包裹管理者」就针对这一点，能将所有要安装的路径全部安排好，当使用者解开有「.rpm」扩展名的文件时，会将当初打包该文件时设定好的路径档案先检查一次，然后依照档案里的设定，将各个文件解开，送到它们应该去的地方；不只如此，它还会制作安装记录，当使用者要移除其中任一个rpm文件时，系统会根据安装记录将该文件反安装，这种做法绝对准确，不会像Windows那样会移除不该拿掉的东西。

Red Hat Linux可以说是相当成功的一个产品，Red Hat公司有「官方版本」（official）供使用者购买，也提供了自由的FTP站供大众直接下载，官方版本与自由下载版本差异在于，官方版本多提供了一些商用软件和印刷精美的说明书。

（2）Slackware http://www.cdrom.com

这是个老字号的门派了，前几年玩Linux的人，几乎都用这套系统。它可完全「手工打造」个人需求的特性，让很多目前已是高手级的玩家仍念念不忘，Slackware在国内用得很多，也许用来做服务器，性能会好些。最新版本安装的过程已改善了不少，各位想要「完全掌控」情况的朋友可以一试。

（3）Debian http://www.debian.org

您通常会在Debian字眼后看见GNU Linux的字样，该派别目前是大家公认的结构最严谨、组织发展最整齐的，它也有一个包裹管理系统称之为「dpk」（Debian
Package），所作的事情和Red Hat的「rpm」异曲同工，使整体文件的管理更加方便。Debian的原始程序代码都是遵循GNU的方式开放的，所以它完全符合开放原代码精神，不像其它的Linux都或多或少的保留了一部份程序代码不开放（Red
Hat是直到6.0版才全部开放的）。最新版本有一部份程序设计师，将Debian移植到其它平台上，所以也可以在其它平台上看见其踪影。

（4）SuSE http://www.suse.com

这是一套在欧洲相当受欢迎的版本，它和XFree86合作开发x86上 的X Server。SuSE安装时可以选择显示德文或英文，它还有自己的一套设定程序叫做「SaX」，可以让使用者较方便的设定，它的安装套件也采用RPM模式，所以要安装、升级与移除程序都非常方便，目前版本8.0。

（5）OpenLinux http://www.caldera.com

这是由Caldera公司推出的版本，并不是很「Open」的一个版本，网上可以下载其Lite版本, 但正式版本是要money的，因为整个套件中有许多商用软件，所以并没有提供网络下载的服务。

（6）TurboLinux http://www.pacific.com

由Pacific HiTech公司发展的套件，该套件在日本市场占有一席之地，从安装到使用接口都是日文的，在国内它与清华大学及研究机构合作研发了中文版本，在国内造成了一股Linux潮流，目前已推出7.0的简体版本，8.0的英文版本。

文件与目录操作

我们一起在风雨中前行

文件与目录操作

用户的数据和程序大多以文件的形式保存。用户使用Linux系统的过程中，需要经常对文件和目录进行操作。本章讲述了文件与目录的基本概念，有关文件和目录操作的命令以及如何在Linux环境下运行DOS命令。

文件与目录的基本概念

文件与文件名

在多数操作系统中都有文件的概念。文件是Linux用来存储信息的基本结构，它是被命名（称为文件名）的存储在某种介质（如磁盘、光盘和磁带等）上的一组信息的集合。Linux文件均为无结构的字符流形式。文件名是文件的标识，它由字母、数字、下划线和圆点组成的字符串来构成。用户应该选择有意义的文件名。Linux要求文件名的长度限制在255个字符以内.为了便于管理和识别，用户可以把扩展名作为文件名的一部分。圆点用于区分文件名和扩展名。扩展名对于将文件分类是十分有用的。用户可能对某些大众已接纳的标准扩展名比较熟悉，例如，C语言编写的源代码文件总是具有C的扩展名。用户可以根据自己的需要，随意加入自己的文件扩展名。

以下例子都是有效的Linux文件名。

preface

chapter1.txt

xu.c

xu.bak

文件的类型

Linux系统中有三种基本的文件类型：普通文件、目录文件和设备文件。

1. 普通文件 普通文件是用户最经常面对的文件。它又分为文本文件和二进制文件。

1)文本文件：这类文件以文本的ASCII码形式存储在计算机中。它是以"行"为基本结构的一种信息组织和存储方式。

2)二进制文件：这类文件以文本的二进制形式存储在计算机中，用户一般不能直接读懂它们，只有通过相应的软件才能将其显示出来。二进制文件一般是可执行程序、图形、图像、声音等等。

2. 目录文件 设计目录文件的主要目的是用于管理和组织系统中的大量文件。它存储一组相关文件的位置、大小等与文件有关的信息。目录文件往往简称为目录。

3. 设备文件 设备文件是Linux系统很重要的一个特色。Linux系统把每一个I/O设备都看成一个文件，与普通文件一样处理，这样可以使文件与设备的操作尽可能统一。从用户的角度来看，对I/O设备的

使用和一般文件的使用一样，不必了解I/O设备的细节。设备文件可以细分为块设备文件和字符设备文件。前者的存取是以一个个字符块为单位的，后者则是以单个字符为单位的。

显示文件内容

使用过DOS命令的人都应该知道，我们可以使用type命令来查看一个文件的内容。在Linux下有五个相关的命令，功能各有千秋，不过它们都象type命令一样，只能用来查看文本文件。

1） cat命令

cat命令是最象type命令的，使用的方法很简单："cat 文件名"。不过比type命令更强大的是，它可以同时查看多个文件："cat 文件名一
文件名二"。

2） more命令

如果文本文件比较长，一屏无法显示完，那么使用cat命令就可能无法看清。这里我们可以简单地使用more来代替cat命令即可。其效果与type
文件名/p类似。使用more命令将一次显示一屏文本，显示满后，停下来，并提示出已显示全部内容的百分比，按空格键就可以看到下一屏。

3） less命令

less命令的功能几乎和more命令一样，也是按页显示文件，不同之处在于less命令在显示文件时允许用户既可以向前又可以向后翻阅文件。

? 向前翻：按b键；

? 向后翻：按p键；

? 指定位置：输入百分比

? 退出：q

我经常使用more和less，几乎不使用cat查看文件。

4） head命令

通过head命令可以仅查看某文件的前几行，格式为：

head 行数 文件名

如果未指定行数，则使用默认值10。

5） tail命令

与head命令相对应的，我们可以使用tail命令来查看文件尾部的内容。通常用来实时监测某个文件是否被修改，通常用来观察日志。如：

tail -f maillog

4. 编辑文件

Linux中有许多文字编辑工具，其中最常用的应该是vi，这是一个广泛应用于所有UNIX系统的编辑器。它的使用有些特别：

首先，可以使用命令"vi 文件名"打开一个文件。

刚启动的时候，vi处于命令状态，不能够输入任何字符。在这个状态下，可以使用方向键进行移动，而需要输入内容时，你需要输入"i"或"a"命令进入编辑状态。编辑完成后，你需要按下"ESC"键回到命令状态。

在命令状态下，你可以输入":q!"不存盘退出，输入":wq"存盘退出。

目录

树型目录结构

在计算机系统中存有大量的文件，如何有效的组织与管理它们，并为用户提供一个使用方便的接口是文件系统的一大任务。Linux系统以文件目录的方式来组织和管理系统中的所有文件。所谓文件目录就是将所有文件的说明信息采用树型结构组织起来--即我们常说的目录。也就是说，整个文件系统有一

个"根"（root），然后在根上分"杈"（directory），任何一个分杈上都可以再分杈，杈上也可以长出"叶子"。"根"和"杈"在Linux中被称为是"目录"或"文件夹"。而"叶子"则是一个个的文件。实践证明，此种结构的文件系统效率比较高。

如前所述，目录也是一种类型的文件。Linux系统通过目录将系统中所有的文件分级、分层组织在一起，形成了Linux文件系统的树型层次结构。以根目录为起点，所有其他的目录都由根目录派生而来。一个典型的Linux系统的树型目录结构如图3.1所示。用户可以浏览整个系统，可以进入任何一个已授权进入的目录，访问那里的文件。

实际上，各个目录结点"之下"都会有一些文件和子目录。并且，系统在建立每一个目录时，都会自动为它设定两个目录文件，一个是"."，代表该目录自己，另一个是".."，代表该目录的父目录，对于根目录，"."和".."都代表其自己。

Linux目录提供了管理文件的一个方便途径。每个目录里面都包含文件。用户可以为自己的文件创建自己的目录，也可以把一个目录下的文件移动或复制到另一目录下，而且能移动整个目录，并且和系统中的其他用户共享目录和文件。也就是说。我们能够方便地从一个目录切换到另一个目录，而且可以设置目录和文件的管理权限，以便允许或拒绝其他人对其进行访问。同时文件目录结构的相互关联性使分享数据变得十分容易，几个用户可以访问同一个文件。因此允许用户设置文件的共享程度。

需要说明的是，根目录是Linux系统中的特殊目录。Linux是一个多用户系统，操作系统本身的驻留程序存放在以根目录开始的专用目录中，有时被指定为系统目录。

安装完Linux后，有许许多多的目录，下面我们就说明一些重要的目录：

/bin：存放着一百多个Linux下常用的命令、工具

/dev：存放着Linux下所有的设备文件！

/home：用户主目录，每建一个用户，就会在这里新建一个与用户同名的目录，给该用户一个自己的空间

/lost+found：顾名思义，一些丢失的文件可能可以在这里找到

/mnt：外部设备的挂接点，通常用cdrom与floppy两个子目录！它的存在简化了光盘与软盘的使用。你只需在塞入光盘后，运行：mount
/mnt/cdrom，就可以将光盘上的内容Mount到/mnt/cdrom上，你就可以访问了。不过你使用完成后，应该离开该目录，并执行umount
/mnt/cdrom。同样的，软盘就是 mount /mnt/floppy和umount /mnt/floppy了。

/proc：这其实是一个假的目录，通过这里你可以访问到内存里的内容。

/sbin：这里存放着系统级的命令与工具

/usr：通常用来安装各种软件的地方

/usr/X11R6 X Window目录

/usr/bin与/usr/sbin 一些后安装的命令与工具

/usr/include、/usr/lib及/usr/share 则是存放一些共享链接库

/usr/local 常用来安装新软件

/usr/src Linux源程序

/boot：Linux就是从这里启动的

/etc：这里存放在Linux大部分的配置文件

/lib：静态链接库

/root：root用户的主目录，这就是特权之一！

/var：通常用来存放一些变化中的东西！

/var/log：存放系统日志

/var/spool：存放一些邮件、新闻、打印队列等

另外，要说明的是，在Linux下“当前目录”、“路径”等概念与Windows下是一样的。

工作目录、用户主目录与路径

为使用户更好地使用目录，我们介绍有关目录的一些基本概念。

I. 工作目录与用户主目录

从逻辑上讲，用户在登录到Linux系统中之后，每时每刻都"处在"某个目录之中，此目录被称作工作目录或当前目录（Working
Directory）。工作目录是可以随时改变的。用户初始登录到系统中时，其主目录（Home Directory）就成为其工作目录。工作目录用"."表示，其父目录用".."表示。

用户主目录是系统管理员增加用户时建立起来的（以后也可以改变），每个用户都有自己的主目录，不同用户的主目录一般互不相同。

用户刚登录到系统中时，其工作目录便是该用户主目录，通常与用户的登录名相同。

用户可以通过一个"~"字符来引用自己的主目录。

例如命令

/home/WANG$ cat ~/class/software_1

和下面的命令

/home/WANG$ cat /home/WANG/class/software_1 意义相同。shell将用用户主目录名来替换"~"字符。

目录层次建立好之后，用户就可以把有关的文件放到相应的目录中，从而实现对文件的组织。

对文件进行访问时，需要用到 "路径"（Path）的概念。

II. 路径

顾名思义，路径是指从树型目录中的某个目录层次到某个文件的一条道路。此路径的主要构成是目录名称，中间用"/"分开。任一文件在文件系统中的位置都是由相应的路径决定的。

用户在对文件进行访问时，要给出文件所在的路径。 路径又分相对路径和绝对路径。 绝对路径是指从"根"开始的路径，也称为完全路径；相对路径是从用户工作目录开始的路径。

应该注意到，在树型目录结构中到某一确定文件的绝对路径和相对路径均只有一条。绝对路径是确定不变的，而相对路径则随着用户工作目录的变化而不断变化。这一点对于我们以后使用某些命令如cp和tar等大有好处。

用户要访问一个文件时，可以通过路径名来引用。并且可以根据要访问的文件与用户工作目录的相对位置来引用它，而不需要列出这个文件的完整的路径名。例如，用户WANG有一个名为class的目录，该目录中有两个文件：software_1和hardware_1。若用户WANG想显示出其class目录中的名为software_1的文件，可以使用下列命令：

/home/WANG$ cat /home/WANG/class/software_1

用户也可以根据文件software_1与当前工作目录的相对位置来引用该文件。这时命令为：

/home/WANG$ cat class/software_1

http://www.linuxsir.com

系列教材三

目录

01.常用命令集一

我们一起在风雨中前行

名称：cal

使用权限：所有使用者
使用方式：cal [-mjy] [month [year]]
说明：
显示日历。若只有一个参数，则代表年份(1-9999)，显示该年的年历。年份必须全部写出：``cal 89\
将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。

1752 年 9 月第 3 日起改用西洋新历，因这时大部份的国家都采用新历，有 10 天被去除，所以该月份的月历有些不同。在此之前为西洋旧历。

参数
-m : 以星期一为每周的第一天方式显示。
-j : 以凯撒历显示，即以一月一日起的天数显示。
-y : 显示今年年历。

范例：
cal : 显示本月的月历。

```
[root@mylinux /root]# date
Tue Aug 15 08:00:18 CST 2000
[root@mylinux /root]# cal
August 2000
```

Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
[root@mylinux /root]#
cal 2001 : 显示公元 2001 年年历。

[root@mylinux /root]# cal 2001
2001
January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1 2 3
7 8 9 10 11 12 13 4 5 6 7 8 9 10 4 5 6 7 8 9 10
14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17
21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24
28 29 30 31 25 26 27 28 25 26 27 28 29 30 31
April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 5 1 2
8 9 10 11 12 13 14 6 7 8 9 10 11 12 3 4 5 6 7 8 9
15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16
22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23
29 30 27 28 29 30 31 24 25 26 27 28 29 30
July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 1
8 9 10 11 12 13 14 5 6 7 8 9 10 11 2 3 4 5 6 7 8
15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15
22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22
29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29
30
October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1
7 8 9 10 11 12 13 4 5 6 7 8 9 10 2 3 4 5 6 7 8
14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15
21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22
28 29 30 31 25 26 27 28 29 30 23 24 25 26 27 28 29
30 31
[root@mylinux /root]#
cal 5 2001 : 显示公元 2001 年 5 月月历。
[root@mylinux /root]# cal 5 2001
May 2001
Su Mo Tu We Th Fr Sa
1 2 3 4 5

6 7 8 9 10 11 12

13 14 15 16 17 18 19

20 21 22 23 24 25 26

27 28 29 30 31

[root@mylinux /root]#

cal -m : 以星期一为每周的第一天方式，显示本月的月历。

[root@mylinux /root]# cal -m

August 2000

Mo Tu We Th Fr Sa Su

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

[root@mylinux /root]#

cal -jy : 以一月一日起的天数显示今年的年历。

[root@mylinux /root]# cal -jy 2000

January February

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

1 32 33 34 35 36

2 3 4 5 6 7 8 37 38 39 40 41 42 43

9 10 11 12 13 14 15 44 45 46 47 48 49 50

16 17 18 19 20 21 22 51 52 53 54 55 56 57

23 24 25 26 27 28 29 58 59 60

30 31

March April

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

61 62 63 64 92

65 66 67 68 69 70 71 93 94 95 96 97 98 99

72 73 74 75 76 77 78 100 101 102 103 104 105 106

79 80 81 82 83 84 85 107 108 109 110 111 112 113

86 87 88 89 90 91 114 115 116 117 118 119 120

121

May June

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

122 123 124 125 126 127 153 154 155

128 129 130 131 132 133 134 156 157 158 159 160 161 162

135 136 137 138 139 140 141 163 164 165 166 167 168 169

142 143 144 145 146 147 148 170 171 172 173 174 175 176

149 150 151 152 177 178 179 180 181 182

July August

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

183 214 215 216 217 218

184 185 186 187 188 189 190 219 220 221 222 223 224 225

191 192 193 194 195 196 197 226 227 228 229 230 231 232

198 199 200 201 202 203 204 233 234 235 236 237 238 239

205 206 207 208 209 210 211 240 241 242 243 244
212 213
September October
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
245 246 275 276 277 278 279 280 281
247 248 249 250 251 252 253 282 283 284 285 286 287 288
254 255 256 257 258 259 260 289 290 291 292 293 294 295
261 262 263 264 265 266 267 296 297 298 299 300 301 302
268 269 270 271 272 273 274 303 304 305
November December
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
306 307 308 309 336 337
310 311 312 313 314 315 316 338 339 340 341 342 343 344
317 318 319 320 321 322 323 345 346 347 348 349 350 351
324 325 326 327 328 329 330 352 353 354 355 356 357 358
331 332 333 334 335 359 360 361 362 363 364 365
366
[root@mylinux /root]#
名称 : crontab
使用权限 : 所有使用者
使用方式 :
crontab [-u user] filecrontab [-u user] { -l | -r | -e }
说明 :
crontab 是用来让使用者在固定时间或固定间隔执行程式之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定 user
的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

参数 :
-e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说
setenv VISUAL joe)
-r : 删除目前的时程表
-l : 列出目前的时程表
时程表的格式如下 :
f1 f2 f3 f4 f5 program
其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program
表示要执行的程式。
当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程式，其余类推
当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推

当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间间隔执行一次，其余类推
当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其余类推

使用者也可以将所有的设定先存放在档案 file 中，用 crontab file 的方式来设定时程表。
例子 :
每月每天每小时的第 0 分钟执行一次 /bin/ls :
0 7 * * * /bin/ls
在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup
:
0 6-12/3 * 12 * /usr/bin/backup

周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata

每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"

20 0-23/2 * * * echo "haha"
注意 :
当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容，若是你不希望收到这样的信，请在每一行空一格之后加上 > /dev/null
2>&1 即可。
名称 : date
使用权限 : 所有使用者
使用方式 :
date [-u] [-d datestr] [-s datestr] [--utc] [--universal]
[--date=datestr] [--set=datestr] [--help] [--version] [+FORMAT] [MMDDhhmm[[CC]YY][.ss]]

说明 :
date 可以用来显示或设定系统的日期与时间，在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中可用的标记列表如下
:
时间方面 :
% : 印出 %
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM
%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数
%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区
日期方面 :
%a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)

%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)
%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)
%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)
若是不以加号作为开头，则表示要设定时间，而时间格式为 MMDDhhmm[[CC]YY][.ss]，其中 MM
为月份，DD 为日，hh 为小时，mm 为分钟，CC 为年份前两位数字，YY 为年份后两位数字，ss 为秒数
参数 :
-d datestr : 显示 datestr 中所设定的时间 (非系统时间)
--help : 显示辅助讯息
-s datestr : 将系统时间设为 datestr 中所设定的时间
-u : 显示目前的格林威治时间
--version : 显示版本编号
例子 :
显示时间后跳行，再显示目前日期 :
date +%T%n%D
显示月份与日数 :
date +%B %d
显示日期与设定时间(12:34:56) :
date --date 12:34:56
注意 :
当你不希望出现无意义的 0 时(比如说 1999/03/07)，则可以在标记中插入 - 符号，比如说 date
+%-H:%-M:%-S 会把时分秒中无意义的 0 给去掉，像是原本的 08:09:04 会变为 8:9:4。另外，只有取得权限者(比如说
root)才能设定系统时间。

当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入 CMOS 中，这样下次重新开机时系统时间才会持续抱持最新的正确值。

名称 : sleep
使用权限 : 所有使用者
使用方式 : sleep [--help] [--version] number[smhd]
说明 : sleep 可以用来将目前动作延迟一段时间
参数说明 :
--help : 显示辅助讯息
--version : 显示版本编号
number : 时间长度，后面可接 s、m、h 或 d

其中 s 为秒，m 为 分钟，h 为小时，d 为日数
例子 :
显示目前时间后延迟 1 分钟，之后再次显示时间 :
date;sleep 1m;date

名称： time
使用权限： 所有使用者
使用方式： time [options] COMMAND [arguments]

说明： time 指令的用途，在于量测特定指令执行时所需消耗的时间及系统资源等资讯。例如 CPU 时间、记忆体、输入输出等等。需要特别注意的是，部分资讯在
Linux 上显示不出来。这是因为在 Linux 上部分资源的分配函式与 time 指令所预设的方式并不相同，以致于 time 指令无法取得这些资料。

把计
-o or --output=FILE
设定结果输出档。这个选项会将 time 的输出写入 所指定的档案中。如果档案已经存在，系统将覆写其内容。
-a or --append
配合 -o 使用，会将结果写到档案的末端，而不会覆盖掉原来的内容。
-f FORMAT or --format=FORMAT
以 FORMAT 字串设定显示方式。当这个选项没有被设定的时候，会用系统预设的格式。不过你可以用环境变数 time 来设定这个格式，如此一来就不必每次登入系统都要设定一次。

一般设定上，你可以用
\t
表示跳栏，或者是用
\n
表示换行。每一项资料要用 % 做为前导。如果要在字串中使用百分比符号，就用 。（学过 C 语言的人大概会觉得很熟悉）
time 指令可以显示的资源有四大项，分别是：
Time resources
Memory resources
IO resources
Command info
详细的内容如下：

Time Resources
E 执行指令所花费的时间，格式是：[hour]:minute:second。请注意这个数字并不代表实际的 CPU 时间。
e 执行指令所花费的时间，单位是秒。请注意这个数字并不代表实际的 CPU 时间。
S 指令执行时在核心模式（kernel mode）所花费的时间，单位是秒。
U 指令执行时在使用者模式（user mode）所花费的时间，单位是秒。
P 执行指令时 CPU 的占用比例。其实这个数字就是核心模式加上使用者模式的 CPU 时间除以总时间。

Memory Resources

M 执行时所占用的实体记忆体的最大值。单位是 KB
t 执行时所占用的实体记忆体的平均值，单位是 KB
K 执行程序所占用的记忆体总量（stack+data+text）的平均大小，单位是 KB
D 执行程序的自有资料区（unshared data area）的平均大小，单位是 KB
p 执行程序的自有堆叠（unshared stack）的平均大小，单位是 KB
X 执行程序间共享内容（shared text）的平均值，单位是 KB
Z 系统记忆体页的大小，单位是 byte。对同一个系统来说这是个常数

IO Resources
F 此程序的主要记忆体页错误发生次数。所谓的主要记忆体页错误是指某一记忆体页已经置换到置换档（swap file)中，而且已经分配给其他程序。此时该页的内容必须从置换档里再读出来。

R 此程序的次要记忆体页错误发生次数。所谓的次要记忆体页错误是指某一记忆体页虽然已经置换到置换档中，但尚未分配给其他程序。此时该页的内容并未被破坏，不必从置换档里读出来

W 此程序被交换到置换档的次数
c 此程序被强迫中断（像是分配到的 CPU 时间耗尽）的次数
w 此程序自愿中断（像是在等待某一个 I/O 执行完毕，像是磁碟读取等等）的次数
I 此程序所输入的档案数
O 此程序所输出的档案数
r 此程序所收到的 Socket Message
s 此程序所送出的 Socket Message
k 此程序所收到的信号 (Signal)数量

Command Info
C 执行时的参数以及指令名称
x 指令的结束代码 (Exit Status)

-p or --portability
这个选项会自动把显示格式设定成为：
real %e
user %U
sys %S
这么做的目的是为了与 POSIX 规格相容。
-v or --verbose
这个选项会把所有程式中用到的资源通通列出来，不但如一般英文语句，还有说明。对不想花时间去熟习格式设定或是刚刚开始接触这个指令的人相当有用。

范例：
利用下面的指令
time -v ps -aux
我们可以获得执行 ps -aux 的结果和所花费的系统资源。如下面所列的资料：
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 1096 472 ? S Apr19 0:04 init
root 2 0.0 0.0 0 0 ? SW Apr19 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Apr19 0:00 [kpiod]
......

root 24269 0.0 1.0 2692 996 pts/3 R 12:16 0:00 ps -aux
Command being timed: "ps -aux"
User time (seconds): 0.05
System time (seconds): 0.06
Percent of CPU this job got: 68%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 0
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 238
Minor (reclaiming a frame) page faults: 46
Voluntary context switches: 0
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0

名称： uptime
使用权限： 所有使用者
使用方式： uptime [-V]
说明： uptime 提供使用者下面的资讯，不需其他参数：
现在的时间
系统开机运转到现在经过的时间
连线的使用者数量
最近一分钟，五分钟和十五分钟的系统负载
参数： -V 显示版本资讯。
范例： uptime
其结果为：
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99
名称：chfn
使用权限：所有使用者
用法：shell>> chfn
说明：提供使用者更改个人资讯，用于 finger and mail username
范例：
shell>> chfn
Changing finger information for user
Password: [del]
Name[]:Johnney Huang ### 提供 finger 时的资料
Office[]:NCCU

Office Phone[]: [del]
Home Phone[]: [del]

名称：chsh
使用权限：所有使用者
用法：shell>> chsh
说明：更改使用者 shell 设定
范例：
shell>> chsh
Changing fihanging shell for user1
Password: [del]
New shell [/bin/tcsh]: ### [是目前使用的 shell]
[del]
shell>> chsh -l ### 展示 /etc/shells 档案内容
/bin/bash
/bin/sh
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh

名称： finger
使用权限： 所有使用者
使用方式： finger [options] user[@address]
说明：finger 可以让使用者查询一些其他使用者的资料。会列出来的资料有：
Login Name
User Name
Home directory
Shell
Login status
mail status
.plan
.project
.forward
其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的 .plan ， .project 和 .forward 等档案里的资料。如果没有就没有。finger 指令并不限定于在同一伺服器上查询，也可以寻找某一个远端伺服器上的使用者。只要给一个像是
E-mail address 一般的地址即可。

参数
-l 多行显示。
-s 单行显示。这个选项只显示登入名称，真实姓名，终端机名称，闲置时间，登入时间，办公室号码及电话号码。如果所查询的使用者是远端伺服器的使用者，这个选项无效。

范例：下列指令可以查询本机管理员的资料：
finger root
其结果如下：

Login: root Name: root
Directory: /root Shell: /bin/bash
Never logged in.
No mail.
No Plan.
名称：last
使用权限：所有使用者
使用方式：shell>> last [options]
说明：显示系统开机以来获是从每月初登入者的讯息
参数
-R 省略 hostname 的栏位
-num 展示前 num 个
username 展示 username 的登入讯息
tty 限制登入讯息包含终端机代号
范例：
shell>> last -R -2
johnney pts/1 Mon Aug 14 20:42 still logged in
johnney pts/0 Mon Aug 14 19:59 still logged in
wtmp begins Tue Aug 1 09:01:10 2000 ### /var/log/wtmp

shell>> last -2 minery
minery pts/0 140.119.217.115 Mon Aug 14 18:37 - 18:40 (00:03)
minery pts/0 140.119.217.115 Mon Aug 14 17:22 - 17:24 (00:02)
wtmp begins Tue Aug 1 09:01:10 2000
名称:login
这个命令都不会就不要干算了！呵呵我也不在这里多费笔墨耽误大家美好青春了^_^
名称：passwd
使用权限：所有使用者
使用方式：passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]
说明：用来更改使用者的密码
参数：
-k
-l
-u
-f
-d 关闭使用者的密码认证功能, 使用者在登入时将可以不用输入密码, 只有具备 root 权限的使用者方可使用.
-S 显示指定使用者的密码认证种类, 只有具备 root 权限的使用者方可使用.
[username] 指定帐号名称.
名称 : who
使用权线 : 所有使用者都可使用
使用方式 : who - [husfV] [user]
说明 : 显示系统中有那些使用者正在上面，显示的资料包含了使用者 ID，使用的终端机，从那边连上来的，上线时间，呆滞时间，CPU 使用量，动作等等。

参数 :
-h : 不要显示标题列

-u : 不要显示使用者的动作/工作
-s : 使用简短的格式来显示
-f : 不要显示使用者的上线位置
-V : 显示程式版本

2005-1-20 11:38:37 上一页 首页 下一页

2005-1-20 上一页 首页 下一页

02.常用命令集二

我们一起在风雨中前行

名称：cat
使用权限：所有使用者
使用方式：cat [-AbeEnstTuv] [--help] [--version] fileName
说明：把档案串连接后传到基本输出（萤幕或加 > fileName 到另一个档案）

参数：
-n 或 --number 由 1 开始对所有输出的行数编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行
-v 或 --show-nonprinting
范例：
cat -n textfile1 > textfile2 把 textfile1 的档案内容加上行号后输入 textfile2 这个档案里

cat -b textfile1 textfile2 >> textfile3 把 textfile1 和 textfile2
的档案内容加上行号（空白行不加）之后将内容附加到 textfile3
名称 : cd
使用权限 : 所有使用者
使用方式 : cd [dirName]
说明 : 变换工作目录至 dirName。 其中 dirName 表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的 home
directory (也就是刚 login 时所在的目录)。
另外，"~" 也表示为 home directory 的意思，"." 则是表示目前所在的目录，".."
则表示目前目录位置的上一层目录。
范例 :
跳到 /usr/bin/ :
cd /usr/bin
跳到自己的 home directory :
cd ~
跳到目前目录的上上两层 :
cd ../..
指令名称 : chmod
使用权限 : 所有使用者 使用方式 : chmod [-cfvR] [--help] [--version]
mode file... 说明 : Linux/Unix 的档案存取权限分为三级 : 档案拥有者、群组、其他。利用

chmod 可以藉以控制档案如何被他人所存取。 参数 :
mode : 权限设定字串，格式如下 : [ugoa...][[+-=][rwxX]...][,...]，其中u
表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。
-c : 若该档案权限确实已经更改，才显示其更改动作
-f : 若该档案权限无法被更改也不要显示错误讯息
-v : 显示权限变更的详细资料
-R : 对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)
--help : 显示辅助说明
--version : 显示版本
范例 :
将档案 file1.txt 设为所有人皆可读取 :
chmod ugo+r file1.txt
将档案 file1.txt 设为所有人皆可读取 :
chmod a+r file1.txt
将档案 file1.txt 与 file2.txt 设为该档案拥有者，与其所属同一个群体者可写入，但其他以外的人则不可写入
:
chmod ug+w,o-w file1.txt file2.txt
将 ex1.py 设定为只有该档案拥有者可以执行 :
chmod u+x ex1.py
将目前目录下的所有档案与子目录皆设为任何人可读取 :
chmod -R a+r *
此外chmod也可以用数字来表示权限如 chmod 777 file
语法为：chmod abc file
其中a,b,c各为一个数字，分别表示User、Group、及Other的权限。
r=4，w=2，x=1
若要rwx属性则4+2+1=7；
若要rw-属性则4+2=6；
若要r-x属性则4+1=7。
范例：
chmod a=rwx file 和 chmod 777 file 效果相同

chmod ug=rwx,o=x file 和 chmod 771 file 效果相同

若用chmod 4755 filename可使此程式具有root的权限

指令名称 : chown
使用权限 : root
使用方式 : chmod [-cfhvR] [--help] [--version] user[:group]
file...
说明 : Linux/Unix 是多人多工作业系统，所有的档案皆有拥有者。利用 chown 可以将档案的拥有者加以改变。一般来说，这个指令只有是由系统管理者(root)所使用，一般使用者没有权限可以改变别人的档案拥有者，也没有权限可以自己的档案拥有者改设为别人。只有系统管理者(root)才有这样的权限。

参数 :
user : 新的档案拥有者的使用者 IDgroup : 新的档案拥有者的使用者群体(group)-c
: 若该档案拥有者确实已经更改，才显示其更改动作-f : 若该档案拥有者无法被更改也不要显示错误讯息-h : 只对于连结(link)进行变更，而非该
link 真正指向的档案-v : 显示拥有者变更的详细资料-R : 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)--help
: 显示辅助说明--version : 显示版本
范例 :
将档案 file1.txt 的拥有者设为 users 群体的使用者 jessie :
chown jessie:users file1.txt
将目前目录下的所有档案与子目录的拥有者皆设为 users 群体的使用者 lamport :
chmod -R lamport:users *

名称：cp
使用权限：所有使用者
使用方式：
cp [options] source dest
cp [options] source... directory
说明：将一个档案拷贝至另一档案，或将数个档案拷贝至另一目录。
参数：
-a 尽可能将档案状态、权限等资料都照原状予以复制。
-r 若 source 中含有目录名，则将目录下之档案亦皆依序拷贝至目的地。
-f 若目的地已经有相同档名的档案存在，则在复制前先予以删除再行复制。
范例：
将档案 aaa 复制(已存在)，并命名为 bbb :
cp aaa bbb
将所有的C语言程式拷贝至 Finished 子目录中 :
cp *.c Finished

名称：cut
使用权限：所有使用者
用法：cut -cnum1-num2 filename
说明：显示每行从开头算起 num1 到 num2 的文字。
范例：
shell>> cat example
test2
this is test1
shell>> cut -c0-6 example ## print 开头算起前 6 个字元
test2
this i

名称 : find
用法 : find
使用说明 :
将档案系统内符合 expression 的档案列出来。你可以指要档案的名称、类别、时间、大小、权限等不同资讯的组合，只有完全相符的才会被列出来。

find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ! 之前的部份为
path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用
-print
为预设 expression
expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。
-mount, -xdev : 只检查和指定目录在同一个档案系统下的档案，避免列出其它档案系统中的档案
-amin n : 在过去 n 分钟内被读取过
-anewer file : 比档案 file 更晚被读取过的档案
-atime n : 在过去 n 天过读取过的档案
-cmin n : 在过去 n 分钟内被修改过
-cnewer file :比档案 file 更新的档案
-ctime n : 在过去 n 天过修改过的档案
-empty : 空的档案-gid n or -group name : gid 是 n 或是 group 名称是 name
-ipath p, -path p : 路径名称符合 p 的档案，ipath 会忽略大小写
-name name, -iname name : 档案名称符合 name 的档案。iname 会忽略大小写
-size n : 档案大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w
是二个位元组。-type
c : 档案类型是 c 的档案。
d: 目录
c: 字型装置档案
b: 区块装置档案
p: 具名贮列
f: 一般档案
l: 符号连结
s: socket
-pid n : process id 是 n 的档案
你可以使用 ( ) 将运算式分隔，并使用下列运算。
exp1 -and exp2
! expr
-not expr
exp1 -or exp2
exp1, exp2
范例:
将目前目录及其子目录下所有延伸档名是 c 的档案列出来。
find . -name "*.c"
将目前目录其其下子目录中所有一般档案列出
find . -ftype f

将目前目录及其子目录下所有最近 20 分钟内更新过的档案列出
find . -ctime -20

名称：less
使用权限：所有使用者
使用方式：
less [Option] filename
说明：

less 的作用与 more 十分相似，都可以用来浏览文字档案的内容，不同的是 less 允许使用者往回卷动以浏览已经看过的部份，同时因为 less 并未在一开始就读入整个档案，因此在遇上大型档案的开启时，会比一般的文书编辑器(如 vi)来的快速。

范例：

名称 : ln
使用权限 : 所有使用者
使用方式 : ln [options] source dist，其中 option 的格式为 :

[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]

[--help] [--version] [--]
说明 : Linux/Unix 档案系统中，有所谓的连结(link)，我们可以将其视为档案的别名，而连结又可分为两种 : 硬连结(hard
link)与软连结(symbolic link)，硬连结的意思是一个档案可以有多个名称，而软连结的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬连结是存在同一个档案系统中，而软连结却可以跨越不同的档案系统。

ln source dist 是产生一个连结(dist)到 source，至于使用硬连结或软链结则由参数决定。
不论是硬连结或软链结都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。
参数：
-f : 链结时先将与 dist 同档名的档案删除-d : 允许系统管理者硬链结自己的目录-i : 在删除与 dist 同档名的档案时先进行询问-n
: 在进行软连结时，将 dist 视为一般的档案-s : 进行软链结(symbolic link)-v : 在连结之前显示其档名-b : 将在链结时会被覆写或删除的档案进行备份-S
SUFFIX : 将备份的档案都加上 SUFFIX 的字尾-V METHOD : 指定备份的方式--help : 显示辅助说明--version
: 显示版本
范例 :
将档案 yy 产生一个 symbolic link : zz
ln -s yy zz
将档案 yy 产生一个 hard link : zz
ln yy xx

名称：locate
使用权限：所有使用者
使用方式： locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-e] [-f] <[-l] [-c]
<[-U] [-u]>
locate [-Vh] [--version] [--help]
说明：
locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的资料库，之后当寻找时就只需查询这个资料库，而不必实际深入档案系统之中了。

在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用

locate your_file_name 的形式就可以了。

参数：
-u
-U 建立资料库，-u 会由根目录开始，-U 则可以指定开始的位置。
-e 将排除在寻找的范围之外。
-l 如果 是 1. 则启动安全模式。在安全模式下，使用者不会看到权限无法看到的档案。这会始速度减慢，因为
locate 必须至实际的档案系统中取得档案的权限资料。
-f 将特定的档案系统排除在外，例如我们没有到理要把 proc 档案系统中的档案放在资料库中。

-q 安静模式，不会显示任何错误讯息。
-n 至多显示 个输出。
-r 使用正规运算式 做寻找的条件。
-o 指定资料库存的名称。
-d 指定资料库的路径
-h 显示辅助讯息
-v 显示更多的讯息
-V 显示程式的版本讯息 范例：
locate chdrv : 寻找所有叫 chdrv 的档案
locate -n 100 a.out : 寻找所有叫 a.out 的档案，但最多只显示 100 个
locate -u : 建立资料库

名称 : ls
使用权限 : 所有使用者
使用方式 : ls [-alrtAFR] [name...]
说明 : 显示指定工作目录下之内容（列出目前工作目录所含之档案及子目录)。
参数：
-a 显示所有档案及目录 (ls内定将档案名或目录名称开头为"."的视为隐藏档，不会列出)
-l 除档案名称外，亦将档案型态、权限、拥有者、档案大小等资讯详细列出
-r 将档案以相反次序显示(原定依英文字母次序)
-t 将档案依建立时间之先后次序列出
-A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)
-F 在列出的档案名称后加一符号；例如可执行档则加 "*", 目录则加 "/"
-R 若目录下有档案，则以下之档案亦皆依序列出
范例：
列出目前工作目录下所有名称是 s 开头的档案，愈新的排愈后面 :
ls -ltr s*
将 /bin 目录以下所有目录及档案详细资料列出 :
ls -lR /bin
列出目前工作目录下所有档案及目录；目录于名称后加 "/", 可执行档于名称后加 "*"
:
ls -AF
名称：more

使用权限：所有使用者
使用方式：more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]
说明：类似 cat ，不过会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h 。

参数：
-num 一次显示的行数
-d 提示使用者，在画面下方显示 [Press space to continue, q to quit.] ，如果使用者按错键，则会显示
[Press h for instructions.] 而不是 哔 声
-l 取消遇见特殊字元 ^L（送纸字元）时会暂停的功能
-f 计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）
-p 不以卷动的方式显示每一页，而是先清除萤幕后再显示内容
-c 跟 -p 相似，不同的是先显示内容再清除其他旧资料
-s 当遇到有连续两行以上的空白行，就代换为一行的空白行
-u 不显示下引号 （根据环境变数 TERM 指定的 terminal 而有所不同）
+/ 在每个档案显示前搜寻该字串（pattern），然后从该字串之后开始显示
+num 从第 num 行开始显示
fileNames 欲显示内容的档案，可为复数个数
范例：
more -s testfile 逐页显示 testfile 之档案内容，如有连续两行以上空白行则以一行空白行显示。
more +20 testfile 从第 20 行开始显示 testfile 之档案内容。

名称：mv
使用权限：所有使用者
使用方式：
mv [options] source dest
mv [options] source... directory
说明：将一个档案移至另一档案，或将数个档案移至另一目录。

参数：
-i 若目的地已有同名档案，则先询问是否覆盖旧档。
范例：

将档案 aaa 更名为 bbb :
mv aaa bbb
将所有的C语言程式移至 Finished 子目录中 :
mv -i *.c

名称：rm
使用权限：所有使用者
使用方式：rm [options] name...
说明：删除档案及目录。
参数：
-i 删除前逐一询问确认。

-f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。
-r 将目录及以下之档案亦逐一删除。
范例：
删除所有C语言程式档；删除前逐一询问确认 :
rm -i *.c
将 Finished 子目录及子目录中所有档案删除 :
rm -r Finished

名称：rmdir
使用权限：于目前目录有适当权限的所有使用者
使用方式： rmdir [-p] dirName
说明： 删除空的目录。

参数：
-p 是当子目录被删除后使它也成为空目录的话，则顺便一并删除。
范例：
将工作目录下，名为 AAA 的子目录删除 :
rmdir AAA
在工作目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为空目录，则 BBB 亦予删除。
rmdir -p BBB/Test

名称：split
使用权限：所有使用者
使用方式：split [OPTION] [INPUT [PREFIX]]
说明：
将一个档案分割成数个。而从 INPUT 分割输出成固定大小的档案，其档名依序为 PREFIXaa, PREFIXab...; PREFIX
预设值为 `x。若没有 INPUT 档或为 `-，则从标准输入读进资料。
参数：
-b, --bytes=SIZE SIZE 值为每一输出档案的大小，单位为 byte。
-C, --line-bytes=SIZE 每一输出档中，单行的最大 byte 数。
-l, --lines=NUMBER NUMBER 值为每一输出档的列数大小。
-NUMBER 与 -l NUMBER 相同。
--verbose 于每个输出档被开启前，列印出侦错资讯到标准错误输出。
--help 显示辅助资讯然后离开。
--version 列出版本资讯然后离开。
SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。
范例：
PostgresSQL 大型资料库备份与回存：
因 Postgres 允许表格大过你系统档案的最大容量，所以要将表格 dump 到单一的档案可能会有问题，使用
split进行档案分割。
% pg_dump dbname | split -b 1m - filename.dump.
重新载入
% createdb dbname
% cat filename.dump.* | pgsql dbname

名称：touch
使用权限：所有使用者
使用方式：
touch [-acfm]
[-r reference-file] [--file=reference-file]
[-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version]
file1 [file2 ...]
说明：
touch 指令改变档案的时间记录。 ls -l 可以显示档案的时间记录。
参数：
a 改变档案的读取时间记录。
m 改变档案的修改时间记录。
c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。
f 不使用，是为了与其他 unix 系统的相容性而保留。
r 使用参考档的时间记录，与 --file 的效果一样。
d 设定时间与日期，可以使用各种不同的格式。
t 设定档案的时间记录，格式与 date 指令相同。
--no-create 不会建立新档案。
--help 列出指令格式。
--version 列出版本讯息。

范例：
最简单的使用方式，将档案的时候记录改为现在的时间。若档案不存在，系统会建立一个新的档案。
touch file
touch file1 file2
将 file 的时间记录改为 5 月 6 日 18 点 3 分，公元两千年。时间的格式可以参考 date 指令，至少需输入
MMDDHHmm ，就是月日时与分。
touch -c -t 05061803 file
touch -c -t 050618032000 file
将 file 的时间记录改变成与 referencefile 一样。
touch -r referencefile file
将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是
24 小时的格式，日期可以使用其他格式如 6 May 2000 。
touch -d "6:03pm" file
touch -d "05/06/2000" file
touch -d "6:03pm 05/06/2000" file
名称 : at
使用权限 : 所有使用者
使用方式 : at -V [-q queue] [-f file] [-mldbv] TIME

说明 : at 可以让使用者指定在 TIME 这个特定时刻执行某个程式或指令，TIME 的格式是
HH:MM其中的 HH 为小时，MM 为分钟，甚至你也可以指定 am, pm, midnight, noon, teatime(就是下午
4

点锺)等口语词。
如果想要指定超过一天内的时间，则可以用 MMDDYY 或者 MM/DD/YY 的格式，其中 MM
是分钟，DD 是第几日，YY 是指年份。另外，使用者甚至也可以使用像是 now + 时间间隔来弹性指定时间，其中的时间间隔可以是 minutes,
hours, days, weeks
另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter
之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作，至于执行的结果将会寄回你的帐号中。
参数:
-V : 印出版本编号
-q : 使用指定的伫列(Queue)来储存，at 的资料是存放在所谓的 queue 中，使用者可以同时使用多个 queue，而 queue
的编号为 a, b, c... z 以及 A, B, ... Z 共 52 个
-m : 即使程式/指令执行完成后没有输出结果, 也要寄封信给使用者
-f file : 读入预先写好的命令档。使用者不一定要使用交谈模式来输入，可以先将所有的指定先写入档案后再一次读入
-l : 列出所有的指定 (使用者也可以直接使用 atq 而不用 at -l)
-d : 删除指定 (使用者也可以直接使用 atrm 而不用 at -d)
-v : 列出所有已经完成但尚未删除的指定
例子 :
三天后的下午 5 点锺执行 /bin/ls :
at 5pm + 3 days /bin/ls
三个星期后的下午 5 点锺执行 /bin/ls :
at 5pm + 2 weeks /bin/ls
明天的 17:20 执行 /bin/date :
at 17:20 tomorrow /bin/date
1999 年的最后一天的最后一分钟印出 the end of world !
at 23:59 12/31/1999 echo the end of world !

03.常用命令集三

我们一起在风雨中前行

名称：/etc/aliases
使用权限：系统管理者
使用方式： 请用 newaliases 更新资料库

说明：
sendmail 会使用一个在 /etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据
aliases档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在
mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫
redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。
/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。
newaliases
的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。范例：
newaliases
下面命令会做相同的事，
sendmail -bi
相关命令:
mail, mailq, newaliases, sendmail

名称：mail
使用权限：所有使用者
使用方式：mail [-iInv] [-s subject] [-c cc-addr] [-b bcc-addr] user1 [user
2 ...]
说明：
mail 不仅只是一个指令， mail 还是一个电子邮件程式，不过利用 mail 来读信的人应该很少吧！对于系统管理者来说 mail 就很有用，因为管理者可以用
mail 写成 script ，定期寄一些备忘录提醒系统的使用者。
参数：
i 忽略 tty 的中断讯号。 (interrupt)
I 强迫设成互动模式。 (Interactive)
v 列印出讯息，例如送信的地点、状态等等。 (verbose)
n 不读入 mail.rc 设定档。
s 邮件标题。
c cc 邮件地址。
b bcc 邮件地址。
范例：
将信件送给一个或以上的电子邮件地址，由于没有加入其他的选项，使用者必须输入标题与信件的内容等。而 user2 没有主机位置，就会送给邮件伺服器的
user2 使用者。
mail user1@email.address
mail user1@email.address user2
将 mail.txt 的内容寄给 user2 同时 cc 给 user1 。如果将这一行指令设成
cronjob 就可以定时将备忘录寄给系统使用者。
mail -s 标题 -c user1 user2 < mail.txt
名称：mesg
使用权限 : 所有使用者
使用方式 : mesg [y|n]
说明 ： 决定是否允许其他人传讯息到自己的终端机介面

参数：
y ： 允许讯息传到终端机介面上。
n ： 不允许讯息传到终端机介面上 。
如果没有设定，则讯息传递与否则由终端机界面目前状态而定。
例子 ：
改变目前讯息设定，改成不允许讯息传到终端机介面上 ：
mesg n

与 mesg 相关的指令有： talk，write，wall。

名称：/etc/aliases
使用权限：系统管理者
使用方式： newaliases
说明：
sendmail 会使用一个在 /etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据
aliases档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在
mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫
redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。

/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。newaliases
的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。

参数：没有任何参数。

范例：
newaliases
下面命令会做相同的事，
sendmail -bi
相关命令：mail, mailq, newaliases, sendmail
名称 : talk
使用权限 : 所有使用者
使用方式 :
talk person [ttyname]
说明 : 与其他使用者对谈
参数：
person : 预备对谈的使用者帐号，如果该使用者在其他机器上，则可输入 person@machine.name

ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息
例子.1 :
与现在机器上的使用者Rollaend对谈，此时 Rollaend 只有一个连线 :
talk Rollaend

接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 `talk jzlee`即可开始对谈，结束请按
ctrl+c

例子.2 :与linuxfab.cx上的使用者Rollaend对谈，使用pts/2来对谈 :
talk Rollaend@linuxfab.cx pts/2
接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 `talk jzlee@jzlee.home`即可开始对谈，结束请按
ctrl+c
注意 : 若萤幕的字会出现不正常的字元，试着按 ctrl+l 更新萤幕画面。

名称 : wall
使用权限 : 所有使用者
使用方式 :
wall [message]
使用说明：
wall 会将讯息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端机介面做为标准传入时，讯息结束时需加上 EOF (通常用
Ctrl+D)
例子 :
传讯息"hi" 给每一个使用者 :
wall hi

名称 : write
使用权限 : 所有使用者
使用方式 :
write user [ttyname]
说明 : 传讯息给其他使用者
参数：
user : 预备传讯息的使用者帐号
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息
例子.1 :
传讯息给 Rollaend，此时 Rollaend 只有一个连线 :
write Rollaend
接下来就是将讯息打上去，结束请按 ctrl+c
例子.2 :传讯息给 Rollaend，Rollaend 的连线有 pts/2，pts/3 :
write Rollaend pts/2
接下来就是将讯息打上去，结束请按 ctrl+c
注意 : 若对方设定 mesg n，则此时讯席将无法传给对方
名称：kill
使用权限：所有使用者
使用方式：
kill [-s signal | -p] [-a] pid ...
kill -l [signal]
说明：kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作，若没有指定，预设是送出终止
(TERM) 的信号

参数：
-s (signal) : 其中可用的讯号有 HUP (1), KILL (9), TERM (15),
分别代表着重跑, 砍掉, 结束; 详细的信号可以用 kill -l
-p : 印出 pid , 并不送出信号
-l (signal) : 列出所有可用的信号名称

范例：
将 pid 为 323 的行程砍掉 (kill) :
kill -9 323
将 pid 为 456 的行程重跑 (restart) :
kill -HUP 456

名称：nice
使用权限：所有使用者
使用方式：nice [-n adjustment] [-adjustment] [--adjustment=adjustment]
[--help] [--version] [command [arg...]]
说明：以更改过的优先序来执行程式, 如果未指定程式, 则会印出目前的排程优先序, 内定的
adjustment
为 10, 范围为 -20 (最高优先序) 到 19 (最低优先序)
参数：
-n adjustment, -adjustment, --adjustment=adjustment
皆为将该原有优先序的增加 adjustment
--help 显示求助讯息
--version 显示版本资讯
范例：
将 ls 的优先序加 1 并执行 :
nice -n 1 ls
将 ls 的优先序加 10 并执行 :
nice ls将 ls 的优先序加 10 并执行
注意 : 优先序 (priority) 为作业系统用来决定 CPU 分配的参数，Linux 使用『回合制(round-robin)』的演算法来做
CPU 排程，优先序越高，所可能获得的 CPU时间就越多。

名称：ps
使用权限：所有使用者
使用方式：ps [options] [--help]
说明：显示瞬间行程 (process) 的动态

参数：
ps 的参数非常多, 在此仅列出几个常用的参数并大略介绍含义
-A 列出所有的行程
-w 显示加宽可以显示较多的资讯
-au 显示较详细的资讯
-aux 显示所有包含其他使用者的行程
au(x) 输出格式 :
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

USER: 行程拥有者
PID: pid
%CPU: 占用的 CPU 使用率
%MEM: 占用的记忆体使用率
VSZ: 占用的虚拟记忆体大小
RSS: 占用的记忆体大小
TTY: 终端的次要装置号码 (minor device number of tty)
STAT: 该行程的状态:
D: 不可中断的静止 (通悸□□缜b进行 I/O 动作)
R: 正在执行中
S: 静止状态
T: 暂停执行
Z: 不存在但暂时无法消除
W: 没有足够的记忆体分页可分配
<: 高优先序的行程
N: 低优先序的行程
L: 有记忆体分页分配并锁在记忆体内 (即时系统或捱A I/O)
START: 行程开始时间
TIME: 执行的时间
COMMAND:所执行的指令
范例：
ps
PID TTY TIME CMD
2791 ttyp0 00:00:00 tcsh
3092 ttyp0 00:00:00 ps
% ps -A
PID TTY TIME CMD
1 ? 00:00:03 init
2 ? 00:00:00 kflushd
3 ? 00:00:00 kpiod
4 ? 00:00:00 kswapd
5 ? 00:00:00 mdrecoveryd
....
% ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.7 1096 472 ? S Sep10 0:03 init [3]
root 2 0.0 0.0 0 0 ? SW Sep10 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Sep10 0:00 [kpiod]
root 4 0.0 0.0 0 0 ? SW Sep10 0:00 [kswapd]
....

名称：pstree
使用权限：所有使用者
使用方式：
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-G|-U] [pid|user]
pstree -V

说明：将所有行程以树状图显示，树状图将会以 pid (如果有指定) 或是以 init 这个基本行程为根 (root)，如果有指定使用者
id ，则树状图会只显示该使用者所拥有的行程

参数：
-a 显示该行程的完整指令及参数，如果是被记忆体置换出去的行程则会加上括号
-c 如果有重覆的行程名，则分开列出（预设值是会在前面加上 *

范例：
pstree
init-+-amd
|-apmd
|-atd
|-httpd---10*[httpd]
%pstree -p
init(1)-+-amd(447)
|-apmd(105)
|-atd(339)
%pstree -c
init-+-amd
|-apmd
|-atd
|-httpd-+-httpd
| |-httpd
| |-httpd
| |-httpd
....

名称：renice
使用权限：所有使用者
使用方式：renice priority [[-p] pid ...] [[-g] pgrp ...]
[[-u] user ...]
说明：重新指定一个或多个行程(Process)的优先序(一个或多个将根据所下的参数而定)
参数：
-p pid 重新指定行程的 id 为 pid 的行程的优先序
-g pgrp 重新指定行程群组(process group)的 id 为 pgrp 的行程（一个或多个）的优先序
-u user 重新指定行程拥有者为 user 的行程的优先序

范例：
将行程 id 为 987 及 32 的行程与行程拥有者为 daemon 及 root 的优先序号码加 1 :
renice +1 987 -u daemon root -p 32
注意 : 每一个行程(Process)都有一个唯一的 (unique) id
名称：top
使用权限：所有使用者
使用方式：top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]

说明：即时显示 process 的动态

参数：
d : 改变显示的更新速度，或是在交谈式指令列(interactive command)按 s

q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称S : 累积模式，会将已完成或消失的子行程 (dead
child process) 的 CPU time 累积起来
s : 安全模式，将交谈式指令取消, 避免潜在的危机
i : 不显示任何闲置 (idle) 或无用 (zombie) 的行程
n : 更新的次数，完成后将会退出 top
b : 批次档模式，搭配 "n" 参数一起使用，可以用来将 top 的结果输出到档案内
范例：
显示更新十次后退出 ;
top -n 10
使用者将不能利用交谈式指令来对行程下命令 :
top -s
将更新显示二次的结果输入到名称为 top.log 的档案里 :
top -n 2 -b < top.log
名称：skill
使用权限：所有使用者
使用方式： skill [signal to send] [options] 选择程序的规则
说明：
送个讯号给正在执行的程序,预设的讯息为 TERM (中断) , 较常使用的讯息为 HUP , INT
, KILL , STOP , CONT ,和 0
讯息有三种写法:分别为 -9 , -SIGKILL , -KILL , 可以使用 -l 或 -L
已列出可使用的讯息。
一般参数：:
-f 快速模式/尚未完成
-i 互动模式/ 每个动作将要被确认
-v 详细输出/ 列出所选择程序的资讯
-w 智能警告讯息/ 尚未完成
-n 没有动作/ 显示程序代号
参数：选择程序的规则可以是, 终端机代号,使用者名称,程序代号,命令名称。
-t 终端机代号 (tty 或 pty)
-u 使用者名称
-p 程序代号 (pid)
-c 命令名称 可使用的讯号:
以下列出已知的讯号名称,讯号代号,功能。
名称 (代号) 功能/ 描述
ALRM 14 离开
HUP 1 离开
INT 2 离开
KILL 9 离开/ 强迫关闭
PIPE 13 离开
POLL 离开
PROF 离开
TERM 15 离开

USR1 离开

USR2 离开

VTALRM 离开

STKFLT 离开/ 只适用于i386, m68k, arm 和 ppc 硬体

UNUSED 离开/ 只适用于i386, m68k, arm 和 ppc 硬体

TSTP 停止 /产生与内容相关的行为

TTIN 停止 /产生与内容相关的行为

TTOU 停止 /产生与内容相关的行为

STOP 停止 /强迫关闭

CONT 从新启动 /如果在停止状态则从新启动,否则忽略

PWR 忽略 /在某些系统中会离开

WINCH 忽略

CHLD 忽略

ABRT 6 核心

FPE 8 核心

ILL 4 核心

QUIT 3 核心

SEGV 11 核心

TRAP 5 核心

SYS 核心 /或许尚未实作

EMT 核心 /或许尚未实作

BUS 核心 /核心失败

XCPU 核心 /核心失败

XFSZ 核心 /核心失败

范例:

停止所有在 PTY 装置上的程序

skill -KILL -v pts/*

停止三个使用者 user1 , user2 , user3

skill -STOP user1 user2 user3

其他相关的命令: kill

04.常用命令集四

我们一起在风雨中前行

名称：expr

使用权限：所有使用者

字串长度

```
shell>> expr length "this is a test"
14
```

数字商数

```
shell>> expr 14 % 9
5
```

从位置处抓取字串

```
shell>> expr substr "this is a test" 3
5
is is
```

数字串 only the first character

```
shell>> expr index "testforthegame" e

2
```

字串真实重现

```
shell>> expr quote thisisatestformela
thisisatestformela
```

名称: tr

1.比方说要把目录下所有的大写档名换为小写档名?

似乎有很多方式，"tr"是其中一种:

```
#!/bin/sh
dir="/tmp/testdir";
files=`find $dir -type f`;
for i in $files
do
dir_name=`dirname $i`;
ori_filename=`basename $i`
new_filename=`echo $ori_filename | tr [:upper:] [:lower:]` > /dev/null;

#echo $new_filename;
mv $dir_name/$ori_filename $dir_name/$new_filename
done
```

2.自己试验中...lowercase to uppercase

```
tr abcdef...[del] ABCDE...[del]
tr a-z A-Z
tr [:lower:] [:upper:]
shell>> echo "this is a test" | tr a-z
A-Z > www
shell>> cat www
THIS IS A TEST
```

3.去掉不想要的字串

```
shell>> tr -d this ### 去掉有关 t.e.s.t
this
man
man
```

test
e
4.取代字串
shell>> tr -s "this" "TEST"

this
TEST
th
TE

指令：clear
用途：清除屏幕用。
使用方法：在 console 上输入 clear。

名称: reset, tset
使用方法: tset [-IQqrs] [-] [-e ch] [-i ch] [-k ch] [-m mapping] [terminal]

使用说明:
reset 其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是
? 的话，这个程式会要求使用者输入终端机的型别。
软驱片 由于这个程式会将终端机设回原始的状态，除了在 login 时使用外，当系统终端机因为程式不正常执行而进入一些奇怪的状态时，你也可以用它来重设终端机o
例如不小心把二进位档用 cat 指令进到终端机，常会有终端机不再回应键盘输入，或是回应一些奇怪字元的问题。此时就可以用 reset 将终端机回复至原始状态。

参数:
-p 将终端机类别显示在萤幕上，但不做设定的动作。这个命令可以用来取得目前终端机的类别。
-e ch 将 erase 字元设成 ch
-i ch 将中断字元设成 ch
-k ch 将删除一行的字元设成 ch
-I 不要做设定的动作，如果没有使用选项 -Q 的话，erase、中断及删除字元的目前值依然会送到萤幕上。
-Q 不要显示 erase、中断及删除字元的值到萤幕上。
-r 将终端机类别印在萤幕上。
-s 将设定 TERM 用的命令用字串的型式送到终端机中，通常在 .login 或 .profile 中用

范例:
让使用者输入一个终端机型别并将终端机设到该型别的预设状态。
reset ?
将 erase 字元设定 control-h
reset -e ^B
将设定用的字串显示在萤幕上
reset -s
Erase is control-B (^B).
Kill is control-U (^U).
Interrupt is control-C (^C).

TERM=xterm;
名称：compress
使用权限：所有使用者
使用方式：compress [-dfvcV] [-b maxbits] [file ...]

说明：
compress 是一个相当古老的 unix 档案压缩指令，压缩后的档案会加上一个 .Z 延伸档名以区别未压缩的档案，压缩后的档案可以以 uncompress
解压。若要将数个档案压成一个压缩档，必须先将档案 tar 起来再压缩。由于 gzip 可以产生更理想的压缩比例，一般人多已改用 gzip 为档案压缩工具。

参数：
c 输出结果至标准输出设备（一般指荧幕）
f 强迫写入档案，若目的档已经存在，则会被覆盖 (force)
v 将程式执行的讯息印在荧幕上 (verbose)
b 设定共同字串数的上限，以位元计算，可以设定的值为 9 至 16 bits 。由于值越大，能使用的共同字串就 越多，压缩比例就越大，所以一般使用预设值
16 bits (bits)
d 将压缩档解压缩
V 列出版本讯息

范例：
将 source.dat 压缩成 source.dat.Z ，若 source.dat.Z 已经存在，内容则会被压缩档覆盖。

compress -f source.dat
将 source.dat 压缩成 source.dat.Z ，并列印出压缩比例。

-v 与 -f 可以一起使用
compress -vf source.dat
将压缩后的资料输出后再导入 target.dat.Z 可以改变压缩档名。
compress -c source.dat > target.dat.Z
-b 的值越大，压缩比例就越大，范围是 9-16 ，预设值是 16 。
compress -b 12 source.dat
将 source.dat.Z 解压成 source.dat ，若档案已经存在，使用者按 y 以确定覆盖档案，若使用
-df 程式则会自动覆盖档案。由于系统会自动加入 .Z 为延伸档名，所以 source.dat 会自动当作
source.dat.Z 处理。

compress -d source.dat
compress -d source.dat.Z
名称： lpd
使用权限： 所有使用者
使用方式：lpd [-l] [#port]
lpd 是一个常驻的印表机管理程式，它会根据 /etc/printcap 的内容来管理本地或远端的印表
机。/etc/printcap 中定义的每一个印表机必须在
/var/lpd 中有一个相对应的目录，目录中以 cf 开头的档案表示一个等待送到适当装置的印表工作。这个档案通常是由 lpr 所产生。
lpr 和 lpd 组成了一个可以离线工作的系统，当你使用 lpr 时，印表机不需要能立即可用，甚至不用

存在。lpd
会自动监视印表机的状况，当印表机上线后，便立即将档案送交处理。这个得所有的应用程式不必等待印表机完成前一工作。
参数：
-1: 将一些除错讯息显示在标准输出上。
#port: 一般而言，lpd 会使用 getservbyname 取得适当的 TCP/IP port，你可以使用这个参数强迫 lpd 使用指定的
port。
范例：
这个程式通常是由 /etc/rc.d 中的程式在系统启始阶段执行。
名称 lpq
作用：
显示列表机贮列中未完成的工作
用法
lpq [l] [P] [user]
说明
lpq 会显示由 lpd 所管理的列表机贮列中未完成的项目。
范例
范例 1. 显示所有在 lp 列表机贮列中的工作
lpq -PlpRank Owner Job Files Total Size1st root 238
(standard input) 1428646 bytes
相关函数
lpr,lpc,lpd
名称： lpr
使用权限： 所有使用者
使用方式：lpr [-P printer]
将档案或是由标准输入送进来的资料送到印表机贮列之中，印表机管理程式 lpd 会在稍后将这个档案送给适当的程式或装置处理。lpr 可以用来将料资送给本地或是远端的主机来处理。

参数：
-p Printer: 将资料送至指定的印表机 Printer，预设值为 lp。
范例：
将 http://www.c和 kkk.c 送到印表机 lp。
lpr -Plp http://www.ckkk.c

名称: lprm
作用：
将一个工作由印表机贮列中移除
用法
/usr/bin/lprm [P] [file...]
说明
尚未完成的印表机工作会被放在印表机贮列之中，这个命令可用来将常未送到印表机的工作取消。由于每一个印表机都有一个独立的贮列，你可以用 -P 这个命令设定想要作用的印列机。如果没有设定的话，会使用系统预设的印表机。
这个命令会检查使用者是否有足够的权限删除指定的档案，一般而言，只有档案的拥有者或是系统管理员才有这个权限。

范例
将印表机 hpprinter 中的第 1123 号工作移除
prm -Phpprinter 1123
将第 1011 号工作由预设印表机中移除
lprm 1011
名称： fdformat
使用权限： 所有使用者
使用方式：fdformat [-n] device
使用说明 : 对指定的软碟机装置进行低阶格式化。使用这个指令对软碟格式化的时候，最好指定像是下面的装置：

/dev/fd0d360 软驱机 A: ，磁片为 360KB 软驱
/dev/fd0h1440 软驱机 A: ，磁片为 1.4MB 软驱
/dev/fd1h1200 软驱机 B: ，磁片为 1.2MB 软驱
如果使用像是 /dev/fd0 之类的装置，如果里面的软驱不是标准容量，格式化可能会失败。在这种情况之下，使用者可以用 setfdprm 指令先行指定必要参数。

参数：
-n 关闭确认功能。这个选项会关闭格式化之后的确认步骤。
范例：
fdformat -n /dev/fd0h1440
软驱机 A 的磁片格式化成 1.4MB 的磁片。并且省略确认的步骤。
名称： mformat
使用权限： 所有使用者
使用方式：
mformat [-t cylinders] [-h heads] [-s sectors] [-l
volume_label] [-F] [-I fsVer-sion] [-S sizecode] [-2 sectors_on_track_0]
[-M software_sector_size] [-a] [-X] [-C] [-H hidden_sectors] [-r root_sectors]
[-B boot_sector] [-0 rate_on_track_0] [-A rate_on_other_tracks] [-1] [-k]
drive:
在已经做过低阶格式化的磁片上建立 DOS 档案系统。如果在编译 mtools 的时候把 USE_2M 的参数打开，部分与
2M 格式相关的参数就会发生作用。否则这些参数（像是 S,2,1,M）不会发生作用。
参数：
-t 磁柱（synlider）数
-h 磁头（head）数
-s 每一磁轨的磁区数
-l 标签
-F 将软驱格式化为 FAT32 格式，不过这个参数还在实验中。
-I 设定 FAT32 中的版本号。这当然也还在实验中。
-S 磁区大小代码，计算方式为 sector = 2^(大小代码+7)
-c 磁丛（cluster）的磁区数。如果所给定的数字会导致磁丛数超过 FAT 表的限制，mformat 会自动放大磁区数。
-s
-M 软体磁区大小。这个数字就是系统回报的磁区大小。通常是和实际的大小相同。
-a 如果加上这个参数，mformat 会产生一组 Atari 系统的序号给这块软碟。
-X 将软碟格式化成 XDF 格式。使用前必须先用 xdfcopy 指令对软碟作低阶格式化的动作。

-C 产生一个可以安装 MS-DOS 档案系统的软驱影像档（disk image）。当然对一个实体软驱机下这个参数是没有意义的。
-H 隐藏磁区的数目。这通常适用在格式化硬碟的分割区时，因为通常一个分割区的前面还有分割表。这个参数未经测试，能不用就不用。
-n 软驱序号
-r 根目录的大小，单位是磁区数。这个参数只对 FAT12 和 FAT16 有效。
-B 使用所指定的档案或是设备的开机磁区做为这片磁片或分割区的开机磁区。当然当中的硬体参数会随之更动。
-k 尽量保持原有的开机磁区。
-0 第 0 轨的资料传输率
-A 第 0 轨以外的资料传输率
-2 使用 2m 格式
-1 不使用 2m 格式
范例：
mformat a:
这样会用预设值把 a: （就是 /dev/fd0）里的软盘格式化。
名称： mkdosfs
使用权限： 所有使用者
使用方式： mkdosfs [-c | -l filename]
[-f number_of_FATs]
[-F FAT_size]
[-i volume_id]
[-m message_file]
[-n volume_name]
[-r root_dir_entry]
[-s sector_per_cluster]
[-v]
device
[block_count]
说明： 建立 DOS 档案系统。 device 指你想要建立 DOS 档案系统的装置代号。像是 /dev/hda1
等等。 block_count 则是你希望配置的区块数。如果 block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。

参数：
-c 建立档案系统之前先检查是否有坏轨。
-l 从得定的档案中读取坏轨记录。
-f 指定档案配置表（FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 档案系统不支援超过
2 个 FAT 表。通常这个不需要改。
-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于软盘，16 位元组用于一般硬碟的分割区，也就是所谓的
FAT16 格式。这个值通常系统会自己选定适当的值。在软盘上用 FAT16 通常不会发生作用，反之在硬碟上用 FAT12 亦然。
-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。
-m 当使用者试图用这片磁片或是分割区开机，而上面没有作业系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用档案编辑好，然后用这个参数指定，或是用

-m -
这样系统会要求你直接输入这段文字。要特别注意的是，档案里的字串长度不要超过 418 个字，包括展开的跳栏符号（TAB）和换行符号（换行符号在
DOS 底下算两个字元！）
-n 指定 Volume Name，就是软驱标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。
-r 指定根目录底下的最大档案数。这里所谓的档案数包括目录。预设值是在软碟上是 112 或是 224 ，在硬碟上是 512。没事不要改这个数字。

-s 每一个磁丛（cluster）的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。
-v 提供额外的讯息
范例：
mkdosfs -n Tester /dev/fd0 将 A 槽里的软盘格式化为 DOS 格式，并将标签设为 Tester

05.Linux备份与压缩命令

我们一起在风雨中前行

Linux备份与压缩命令

用户经常需要备份计算机系统中的数据，为了节省存储空间，常常将备份文件进行压缩。下面分别介绍备份与压缩的命令。

tar命令

作用：tar可以为文件和目录创建档案。利用tar，用户可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件。tar最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案，如软盘。利用tar命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。Linux上的tar是GNU版本的。

语法：tar [主选项+辅选项] 文件或者目录

使用该命令时，主选项是必须要有的，它告诉tar要做什么事情，辅选项是辅助使用的，可以选用。

参数：

c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。

r 把要存档的文件追加到档案文件的未尾。例如用户已经作好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。

t 列出档案文件的内容，查看已经备份了哪些文件。

u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。

x 从档案文件中释放文件。

辅助选项：

b 该选项是为磁带机设定的。其后跟一数字，用来说明区块的大小，系统预设值为20（20*512 bytes）。

f 使用档案文件或设备，这个选项通常是必选的。

k 保存已经存在的文件。例如我们把某个文件还原，在还原的过程中，遇到相同的文件，不会进行覆盖。

m 在还原文件时，把所有文件的修改时间设定为现在。

M 创建多卷的档案文件，以便在几个磁盘中存放。

v 详细报告tar处理的文件信息。如无此选项，tar不报告文件信息。

w 每一步都要求确认。

z 用gzip来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，但还原时也一定要使用该选项进行解压缩。

例子：

例1：把/home目录下包括它的子目录全部做备份文件，备份文件名为usr.tar。

$ tar cvf usr.tar /home

例2：把/home目录下包括它的子目录全部做备份文件，并进行压缩，备份文件名为usr.tar.gz 。

$ tar czvf usr.tar.gz /home

例3：把usr.tar.gz这个备份文件还原并解压缩。

$ tar xzvf usr.tar.gz

例4：查看usr.tar备份文件的内容，并以分屏方式显示在显示器上。

$ tar tvf usr.tar | more

要将文件备份到一个特定的设备，只需把设备名作为备份文件名。

例5：用户在/dev/fd0设备的软盘中创建一个备份文件，并将/home 目录中所有的文件都拷贝到备份文件中。

$ tar cf /dev/fd0 /home

要恢复设备磁盘中的文件，可使用xf选项：

$ tar xf /dev/fd0

如果用户备份的文件大小超过设备可用的存贮空间，如软盘，您可以创建一个多卷的tar备份文件。M选项指示tar命令提示您使用一个新的存贮设备，当使用M选项向一个软驱进行存档时，tar命令在一张软盘已满的时候会提醒您再放入一张新的软盘。这样您就可以把tar档案存入几张磁盘中。

$ tar cMf /dev/fd0 /home

要恢复几张盘中的档案，只要将第一张放入软驱，然后输入有x和M选项的tar命令。在必要时您会被提醒放入另外一张软盘。

$ tar xMf /dev/fd0

gzip命令

作用：减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip是在Linux系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。

语法：gzip [选项] 压缩（解压缩）的文件名

参数：

-c 将输出写到标准输出上，并保留原有文件。

-d 将压缩文件解压。

-l 对每个压缩文件，显示下列字段：

压缩文件的大小

未压缩文件的大小

压缩比

未压缩文件的名字

-r 递归式地查找指定目录并压缩其中的所有文件或者是解压缩。

-t 测试，检查压缩文件是否完整。

-v 对每一个压缩和解压的文件，显示文件名和压缩比。

-num 用指定的数字num调整压缩的速度，-1或--fast表示最快压缩方法（低压缩比），-9或--best表示最慢压缩方法（高压缩比）。系统缺省值为6。

假设一个目录/home下有文件mm.txt、sort.txt、xx.com。

例子：

例1：把/home目录下的每个文件压缩成.gz文件。

$ cd /home

$ gzip *

$ ls

m.txt.gz sort.txt.gz xx.com.gz

例2：把例1中每个压缩的文件解压，并列出详细的信息。

$ gzip -dv *

mm.txt.gz 43.1%-----replaced with mm.txt

sort.txt.gz 43.1%-----replaced with sort.txt

xx.com.gz 43.1%-----replaced with xx.com

$ ls

mm.txt sort.txt xx.com

例3：详细显示例1中每个压缩的文件的信息，并不解压。

$ gzip -l *

compressed uncompr. ratio uncompressed_name

277 445 43.1% mm.txt

278 445 43.1% sort.txt

277 445 43.1% xx.com

$ ls

mm.txt.gz sort.txt.gz xx.com.gz

例4：压缩一个tar备份文件，如usr.tar，此时压缩文件的扩展名为.tar.gz

$ gzip usr.tar

$ ls

usr.tar.gz

unzip命令

作用：用MS Windows下的压缩软件winzip压缩的文件如何在Linux系统下展开呢？可以用unzip命令，该命令用于解扩展名为.zip的压缩文件。

语法：unzip [选项] 压缩文件名.zip

参数：

-x 文件列表 解压缩文件，但不包括指定的file文件。

-v 查看压缩文件目录，但不解压。
-t 测试文件有无损坏，但不解压。
-d 目录 把压缩文件解到指定目录下。
-z 只显示压缩文件的注解。
-n 不覆盖已经存在的文件。
-o 覆盖已存在的文件且不要求用户确认。
-j 不重建文档的目录结构，把所有文件解压到同一目录下。

例子：
例1：将压缩文件text.zip在当前目录下解压缩。
$ unzip text.zip
例2：将压缩文件text.zip在指定目录/tmp下解压缩，如果已有相同的文件存在，要求unzip命令不覆盖原先的文件。
$ unzip -n text.zip -d /tmp
例3：查看压缩文件目录，但不解压。
$ unzip -v text.zip
zgrep命令
这个命令的功能是在压缩文件中寻找匹配的正则表达式，用法和grep命令一样，只不过操作的对象是压缩文件。如果用户想看看在某个压缩文件中有没有某一句话，便可用zgrep命令。

http://www.linuxsir.com

06.在Linux环境下运行DOS命令

我们一起在风雨中前行

在Linux环境下运行DOS命令

Linux系统提供了一组称为mtools的可移植工具，可以让用户轻松地从标准的DOS软盘上读、写文件和目录。它们对DOS和Linux环境之间交换文件非常有用。它们是不具备共同的文件系统格式的系统之间交换文件的有力手段。对于一个MS-DOS的软盘，只要把软盘放在软驱中，就可以利用mtools提供的命令来访问软盘上的文件。

mtools的主要命令如下：
mcd 目录名 改变MSDOS目录；

mcopy 源文件 目标文件 在MSDOS和Unix之间复制文件；

mdel 文件名 删除MSDOS文件；

mdir 目录名 显示MSDOS目录；

mformat 驱动器号 在低级格式化的软盘上创建MSDOS文件系统；

rnlabel 驱动器号 产生MSDOS卷标；

mmd 目录名 建立MSDOS目录；

mrd 目录名 删除MSDOS目录；

mren 源文件 目标文件 重新命名已存在的MSDOS文件；

mtype 文件名 显示MSDOS文件的内容。

这些命令和对应的不加 m的 MSDOS命令非常相似。

例1：在Linux环境下看DOS盘最上层的目录的内容：

$ mdir a:

Volume in drive A has no label

Volume Serial Number is 15F6-3362

Directory of A:\

SS6 CPP 331 09-24-99 7:41 ss6.cpp

CH9

11-20-99 16:22 ch9

XXQ 0 11-20-99 16:24 xxq

95CZXTA DOC 36,864 06-15-98 22:51 95czxta.doc

95CZXTB DOC 39,936 06-16-98 7:18 95czxtb.doc

HTCA DOC 27,136 01-08-99 0:13 htca.doc

HTCB DOC 27,136 01-08-99 0:12 htcb.doc

6 file（s） 131,403 bytes

1 dir（s） 1,295,872 bytes free

例2：将DOS盘上的文件xxq复制到当前目录下，并用ls命令进行验证。

$ mcopy a:\htca.doc

$ ls －l htca.doc

-rw-r—— -r- - 1 xxq xxq 27136 Jan 1 01:80 htca.doc

http://www.linuxsir.com

07.Linux文件内容查询命令

我们一起在风雨中前行

Linux文件内容查询命令

grep、fgrep和egrep命令

这组命令以指定模式搜索文件，并通知用户在什么文件中搜索到与指定的模式匹配的字符串，并打印出所有包含该字符串的文本行，在该文本行的最前面是该行所在的文件名。grep命令一次只能搜索一个指定的模式；egrep命令检索扩展的正则表达式（包括表达式组和可选项）；fgrep命令检索固定字符串，它不识别正则表达式，是快速搜索命令。

这组命令在搜索与定位文件中特定的主题方面非常有用。要搜索的模式可以被认为是一些关键词，您可以用它们来搜索文件中包含的这些关键词。编写程序时，可以用它来寻找某一个函数，或是相关的词组。grep命令的搜索功能比fgrep强大，因为grep命令的搜索模式可以是正则表达式，而fgrep却不能。有关正则表达式请参见shell一章。

该组命令中的每一个命令都有一组选项，利用这些选项可以改变其输出方式。例如，可以在搜索到的文本行上加入行号，或者只输出文本行的行号，或者输出所有与搜索模式不匹配的文本行，或只简单地输出已搜索到指定模式的文件名，并且可以指定在查找模式时忽略大小写。

这组命令在指定的输入文件中查找与模式匹配的行。如果没有指定文件，则从标准输入中读取。正常情况下，每个匹配的行被显示到标准输出。如果要查找的文件是多个，则在每一行输出之前加上文件名。

语法：

grep [选项] [查找模式] [文件名1，文件名2，……]

egrep [选项] [查找模式] [文件名1，文件名2，……]

fgrep [选项] [查找模式] [文件名1，文件名2，……]

参数：

- E 每个模式作为一个扩展的正则表达式对待。

- F 每个模式作为一组固定字符串对待（以新行分隔），而不作为正则表达式。

- b在输出的每一行前显示包含匹配字符串的行在文件中的字节偏移量。

- c 只显示匹配行的数量。

- i 比较时不区分大小写。

- h 在查找多个文件时，指示grep不要将文件名加入到输出之前。

- l 显示首次匹配串所在的文件名并用换行符将其隔开。当在某文件中多次出现匹配串时，不重复显示此文件名。

- n 在输出前加上匹配串所在行的行号（文件首行行号为1）。

- v 只显示不包含匹配串的行。

- x 只显示整行严格匹配的行。

- e expression 指定检索使用的模式。用于防止以“-”开头的模式被解释为命令选项。

- f expfile 从expfile文件中获取要搜索的模式，一个模式占一行。

对该组命令的使用还需注意以下方面：

在命令后键入搜索的模式，再键入要搜索的文件。其中，文件名列表中也可以使用特殊字符，如“*”等，用来生成文件名列表。如果想在搜索的模式中包含有空格的字符串，可以用单引号把要搜索的模式括起来，用来表明搜索的模式是由包含空格的字符串组成。否则，Shell将把空格认为是命令行参数的定界符，而grep命令将把搜索模式中的单词解释为文件名列表中的一部分。在下面的例子中，grep命令在文件example中搜索模式“text
file”。

$ grep ’text file’ example

用户可以在命令行上用Shell特殊字符来生成将要搜索的文件名列表。在下面的例子中，特殊字符

“*”用来生成一个文件名列表，该列表包含当前目录下所有的文件。该命令将搜索出当前目录下所有文件中与模式匹配的行。

$ grep data *

特殊字符在搜索一组指定的文件时非常有用。例如，如果想搜索所有的C程序源文件中特定的模式，您可以用“*.c”来指定文件名列表。假设用户的
C程序中包含一些不必要的转向语句（goto语句），想要找到这些语句，可以用如下的命令来搜索并显示所有包含goto语句的代码行：

$ grep goto *.c

用户可以在命令行上键入搜索模式，也可以使用-f选项从指定文件中读取要搜索的模式。在文件中，每个搜索模式占一行。如果经常要搜索一组常见字符串时，这个功能非常有用。在下面的例子中，用户要在文件exam中搜索字符串“editor”和“create”，就把要搜索的模式放置在文件mypats中，然后，grep命令从文件mypats中读取要搜索的模式。

$ cat mypats
editor
create
$ grep -f mypats exam

文件查找命令

find命令

功能：在目录结构中搜索文件，并执行指定的操作。此命令提供了相当多的查找条件，功能很强大。

语法：find 起始目录 寻找条件 操作

说明：find命令从指定的起始目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件并对之采取相关的操作。

该命令提供的寻找条件可以是一个用逻辑运算符not、and、or组成的复合条件。逻辑运算符and、or、not的含义为：

（1）and：逻辑与，在命令中用“-a”表示，是系统缺省的选项，表示只有当所给的条件都满足时，寻找条件才算满足。例如：

$ find －name ’tmp’ －xtype c -user ’inin’

该命令寻找三个给定条件都满足的所有文件。

（2）or：逻辑或，在命令中用“-o”表示。该运算符表示只要所给的条件中有一个满足时，寻找条件就算满足。

例如：

$ find －name ’tmp’ －o －name ’mina*’

该命令查询文件名为’tmp’或是匹配’mina*’的所有文件。

（3）not：逻辑非，在命令中用“！”表示。该运算符表示查找不满足所给条件的文件。

例如：

$ find ! －name ’tmp’

该命令查询文件名不是’tmp’的所有文件。

需要说明的是：当使用很多的逻辑选项时，可以用括号把这些选项括起来。为了避免Shell本身对括号引起误解，在话号前需要加转义字符“\\”来去除括号的意义。

例：$ find \\（－name ’tmp’ －xtype c -user ’inin’ \\） 寻找条件有以下选项：

首先，下列各个选项中的n值可以有三种输入方式，假设n为20，则：

+20 表示20以后（21，22，23等）

-20 表示20以前（19，18，17等）

20 表示正好是20

1. 以名称和文件属性查找。

- name ’字串’ 查找文件名匹配所给字串的所有文件，字串内可用通配符*、?、[ ]。

- lname ’字串’查找文件名匹配所给字串的所有符号链接文件，字串内可用通配符*、?、[ ]。

-gid n 查找属于ID号为n的用户组的所有文件。

-uid n 查找属于ID号为n的用户的所有文件。

-group ’字串’ 查找属于用户组名为所给字串的所有的文件。

-user ’字串’ 查找属于用户名为所给字串的所有的文件。

-empty 查找大小为0的目录或文件。

-path ’字串’ 查找路径名匹配所给字串的所有文件，字串内可用通配符*、?、[ ]。

-perm 权限 查找具有指定权限的文件和目录，权限的表示可以如711，644。

-size n[bckw] 查找指定文件大小的文件，n后面的字符表示单位，缺省为b，代表512字节的块。

-type x 查找类型为x的文件，x为下列字符之一：

b 块设备文件

c 字符设备文件

d 目录文件

p 命名管道（FIFO）

f 普通文件

l 符号链接文件（symbolic links）

s socket文件

-xtype x 与-type基本相同，但只查找符号链接文件。

2. 以时间为条件查找

- amin n 查找n分钟以前被访问过的所有文件。

- atime n 查找n天以前被访问过的所有文件。

- cmin n 查找n分钟以前文件状态被修改过的所有文件。

- ctime n 查找n天以前文件状态被修改过的所有文件。

- mmin n 查找n分钟以前文件内容被修改过的所有文件。

- mtime n 查找n天以前文件内容被修改过的所有文件。

3. 可执行的操作

- exec 命令名称 {} 对符合条件的文件执行所给的Linux 命令，而不询问用户是否需要执行该命令。{}表示命令的参数即为所找到的文件；命令的末尾必须以“
\\；”结束。

- ok 命令名称 { } 对符合条件的文件执行所给的Linux 命令，与exec不同的是，它会询问用户是否需要执行该命令。

- ls 详细列出所找到的所有文件。

- fprintf 文件名 将找到的文件名写入指定文件。

- print 在标准输出设备上显示查找出的文件名。

- printf 格式 格式的写法请参考有关C语言的书。

例1：查找当前目录中所有以main开头的文件，并显示这些文件的内容。

$ find . - name ‘main*’ - exec more {} \\;

例2：删除当前目录下所有一周之内没有被访问过的a .out或*.o文件。

$ find . \\ (- name a.out - o - name ‘*.o’ \\) \\

> - atime +7 - exec rm {} \\;

说明如下： 命令中的“.”表示当前目录，此时find将从当前目录开始，逐个在其子目录中查找满足后面指定条件的文件。\\（和\\）表示括号（），其中的“\\”称为转义符。之所以这样写是由于对

Shell而言，（和）另有不同的含义，而不是这里的用于组合条件的用途。“-
name a.out”是指要查找名为a.out的文件；“- name ‘*.o’”是指要查找所有名字以 .o结尾的文件。这两个- name之间的-
o表示逻辑或（or），即查找名字为a.out或名字以 .o结尾的文件，find在当前目录及其子目录下找到这佯的文件之后，再进行判断，看其最后访问时间是否在7天以前（条件－atime
+7），若是，则对该文件执行命令rm（- exec rm{ }\\;）。其中{ }代表当前查到的符合条件的文件名，\\;则是语法所要求的。上述命令中第一行的最后一个\\是续行符。当命令太长而在一行写不下时，可输入一个\\，之后系统将显示一个>，指示用户继续输入命令。

locate命令

功能：locate命令用于查找文件，它比find命令的搜索速度快，它需要一个数据库，这个数据库由每天的例行工作（crontab）程序来建立。当我们建立好这个数据库后，就可以方便地来搜寻所需文件了。

语法： locate 相关字

例如：查找相关字issue

```
$ locate issue
/etc/issue
/etc/issue.net
/usr/man/man5/issue.5
/usr/man/man5/issue.net.5 。
```

http://www.linuxsir.com

08.Linux文件的复制、删除和移动命令

我们一起在风雨中前行

Linux文件的复制、删除和移动命令

cp命令

功能：将给出的文件或目录拷贝到另一文件或目录中，就如同DOS下的copy命令一样，功能非常强大。

语法：cp [选项] 源文件或目录 目标文件或目录

说明：该命令把指定的源文件复制到目标文件或把多个源文件复制到目标目录中。

参数：

- a 该选项通常在拷贝目录时使用。它保留链接、文件属性，并递归地拷贝目录，其作用等于dpR选项的组合。

- d 拷贝时保留链接。

- f 删除已经存在的目标文件而不提示。

- i 和f选项相反，在覆盖目标文件之前将给出提示要求用户确认。回答y时目标文件将被覆盖，是交互式拷贝。

- p 此时cp除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中。

- r 若给出的源文件是一目录文件，此时cp将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。

- l 不作拷贝，只是链接文件。

需要说明的是，为防止用户在不经意的情况下用cp命令破坏另一个文件，如用户指定的目标文件名是一个已存在的文件名，用cp命令拷贝文件后，这个文件就会被新拷贝的源文件覆盖，因此，建议用户在使用cp命令拷贝文件时，最好使用i选项。

$ cp - i exam1.c /usr/wang/shiyan1.c

该命令将文件exam1.c拷贝到/usr/wang 这个目录下，并改名为 shiyan1.c。若不希望重新命名，可以使用下面的命令：

$ cp exam1.c /usr/ wang/

$ cp - r /usr/xu/ /usr/liu/ 将/usr/xu目录中的所有文件及其子目录拷贝到目录/usr/liu中。

mv命令

功能：为文件或目录改名或将文件由一个目录移入另一个目录中。该命令如同DOS下的ren和move的组合。

语法：mv [选项] 源文件或目录 目标文件或目录

说明：视mv命令中第二个参数类型的不同（是目标文件还是目标目录），mv命令将文件重命名或将其移至一个新的目录中。当第二个参数类型是文件时，mv命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv命令将各参数指定的源文件均移至目标目录中。在跨文件系统移动文件时，mv先拷贝，再将原有文件删除，而链至该文件的链接也将丢失。

参数：

- I 交互方式操作。如果mv操作将导致对已存在的目标文件的覆盖，此时系统询问是否重写，要求用户回答y或n，这样可以避免误覆盖文件。

- f 禁止交互操作。在mv操作要覆盖某已有的目标文件时不给任何指示，指定此选项后，i选项将不再起作用。

如果所给目标文件（不是目录）已存在，此时该文件的内容将被新文件覆盖。为防止用户在不经意的情况下用mv命令破坏另一个文件，建议用户在使用mv命令移动文件时，最好使用i选项。

需要注意的是，mv与cp的结果不同。mv好象文件“搬家”，文件个数并末增加，而cp对文件进行复制，文件个数增加了。

例1：将/usr/xu中的所有文件移到当前目录（用“.”表示）中：

$ mv /usr/xu/ * .

例2：将文件wch.txt重命名为wjz.doc

$ mv wch.txt wjz.doc

rm命令

功能：在linux中创建文件很容易，系统中随时会有文件变得过时且毫无用处。用户可以用rm命令将其删除。该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

语法： rm [选项] 文件…

如果没有使用- r选项，则rm不会删除目录。

参数：

- f 忽略不存在的文件，从不给出提示。

- r 指示rm将参数中列出的全部目录和子目录均递归地删除。

- i 进行交互式删除。

使用rm命令要格外小心。因为一旦一个文件被删除，它是不能被恢复的。例如，用户在输入cp，mv或其他命令时，不小心误输入了rm命令，当用户按了回车键并认识到自己的错误时，已经太晚了，文件已经没有了。为了防止此种情况的发生，可以使用rm命令中的
i选项来确认要删除的每个文件。如果用户输入y，文件将被删除。如果输入任何其他东西，文件将被保留。在下一个例子中，用户要删除文件test和example。然后会被要求对每个文件进行确认。用户最终决定删除example文件，保留test文件。

$ rm - ii test example Remove test ?n

Remove example ?y

http://www.linuxsir.com

09.Linux目录的创建与删除命令

我们一起在风雨中前行

Linux目录的创建与删除命令

mkdir命令

功能：创建一个目录（类似DOS下的md命令）。

语法：mkdir [选项] dirname

说明：该命令创建由dirname命名的目录。要求创建目录的用户在当前目录中 （dirname的父目录中）具有写权限，并且dirname不能是当前目录中已有的目录或
文件名称。

参数：

- m 对新建目录设置存取权限。也可以用chmod命令设置。

- p 可以是一个路径名称。此时若路径中的某些目录尚不存在， 加上此选项后， 系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录。

例如：在当前目录中建立inin 和inin下的/mail目录，也就是连续建两个目录。

$ mkdir - p - m 700 ./inin/mail/

该命令的执行结果是在当前目录中创建嵌套的目录层次inin /mail，权限设置为只 有文件主有读、写和执行权限。

rmdir 命令

功能：删除空目录。

语法：rmdir [选项] dirname

说明：dirname表示目录名。该命令从一个目录中删除一个或多个子目录项。需要特别注意的是，一个目录被删除之前必须是空的。（注意，rm - r dir命令可代替 rmdir，但是有很大危险性。）删除某目录时也必须具有对父目录的写权限。

参数：

- p 递归删除目录dirname，当子目录删除后其父目录为空时，也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息。

例如：

$ rmdir - p /usr/xu/txt 将/usr/xu/txt目录删除。

改变工作目录、显示目录内容命令

cd 命令

功能：改变工作目录。

语法：cd [directory]

说明：该命令将当前目录改变至directory所指定的目录。若没有指定directory，则回到用户的主目录。为了改变到指定目录，用户必须拥有对指定目录的执行和读 权限。

该命令可以使用通配符（通配符含义请参见第十章）。

例如：假设用户当前目录是：/home/xu，现需要更换到/home/xu/pro目录中，

$ cd pro 此时，用户可以执行pwd命令来显示工作目录。

$ pwd /home/xu/pro

pwd 命令

功能：在Linux层次目录结构中，用户可以在被授权的任意目录下利用mkdir命令创建新目录，也可以利用cd命令从一个目录转换到另一个目录。然而，没有提示符来告知用户目前处于哪一个目录中。要想知道当前所处的目录，可以使用pwd命令，该命令显示整个路径名。

语法：pwd

说明：此命令显示出当前工作目录的绝对路径。

例：

$ pwd /home/xu 显示的路径名为/home/xu，每个目录名都用“/”隔开，根目录以开头的“/”表示。

ls 命令

功能：ls是英文单词list的简写，其功能为列出目录的内容。这是用户最常用的一个命令之一，因为用户需要不时地查看某个目录的内容。该命令类似于DOS下的dir命令。

语法：ls [选项] [目录或是文件]

对于每个目录，该命令将列出其中的所有子目录与文件。对于每个文件，ls将输出 其文件名以及所要求的其他信息。默认情况下，输出条目按字母顺序排序。当未给出目录名或是文件名时，就显示当前目录的信息。

参数：

- a 显示指定目录下所有子目录与文件，包括隐藏文件。

- A 显示指定目录下所有子目录与文件，包括隐藏文件。但不列出“.”和 “..”。

- b 对文件名中的不可显示字符用八进制逃逸字符显示。

- c 按文件的修改时间排序。

- C 分成多列显示各项。

- d 如果参数是目录，只显示其名称而不显示其下的各文件。往往与l选项一起使 用，以得到目录的详细信息。

- f 不排序。该选项将使lts选项失效，并使aU选项有效。

- F 在目录名后面标记“/”，可执行文件后面标记“*”，符号链接后面标记 “@”，管道（或FIFO）后面标记“|”，socket文件后面标记“=”。

- i 在输出的第一列显示文件的i节点号。

- l 以长格式来显示文件的详细信息。这个选项最常用。

每行列出的信息依次是： 文件类型与权限 链接数 文件属主 文件属组 文件大小 建立或最近修改的时间 名字

对于符号链接文件，显示的文件名之后有“—〉”和引用文件路径名。

对于设备文件，其“文件大小”字段显示主、次设备号，而不是文件大小。

目录中的总块数显示在长格式列表的开头，其中包含间接块。

- L 若指定的名称为一个符号链接文件，则显示链接所指向的文件。

- m 输出按字符流格式，文件跨页显示，以逗号分开。

- n 输出格式与l选项相同，只不过在输出中文件属主和属组是用相应的UID号和 GID号来表示，而不是实际的名称。

- o 与l选项相同，只是不显示拥有者信息。

- p 在目录后面加一个“/”。

- q 将文件名中的不可显示字符用“?”代替。

- r 按字母逆序或最早优先的顺序显示输出结果。

- R 递归式地显示指定目录的各个子目录中的文件。

- s 给出每个目录项所用的块数，包括间接块。

- t 显示时按修改时间（最近优先）而不是按名字排序。若文件修改时间相同，则 按字典顺序。修改时间取决于是否使用了c或u选项。缺省的时间标记是最后一次修
改时间。

- u 显示时按文件上次存取的时间（最近优先）而不是按名字排序。即将-t的时间 标记修改为最后一次访问的时间。

- x 按行显示出各排序项的信息。

用ls - l命令显示的信息中，开头是由10个字符构成的字符串，其中第一个字符表示文件类型，它可以是下述类型之一：

- 普通文件

d 目录

l 符号链接

b 块设备文件

c 字符设备文件

后面的9个字符表示文件的访问权限，分为3组，每组3位。

第一组表示文件属主的权限，第二组表示同组用户的权限，第三组表示其他用户的权限。每一组的三个字 符分别表示对文件的读、写和执行权限。

各权限如下所示：

r 读

w 写

x 执行。对于目录，表示进入权限。

s 当文件被执行时，把该文件的UID或GID赋予执行进程的UID（用户ID）或GID（组 ID）。

t 设置标志位（留在内存，不被换出）。如果该文件是目录，在该目录中的文件只能被超级用户、目录拥有者或文件属主删除。如果它是可执行文件，在该文件执行
后，指向其正文段的指针仍留在内存。这样再次执行它时，系统就能更快地装入该文件。

- 没有设置权限。

例1：列出当前目录的内容。

$ ls

-F bin/ lib/ var/

etc/ tmp/ index. html

例2：列出某个目录的内容。

$ ls －F /home/xu Mail/

maple/ mm/ ptr/ telnetd*

fd/ mbox mount sobsrc. tgz tmp/

例3：列出某个目录下所有的文件（包括隐藏文件）。

$ ls -aF /home/xu

./ . .elm/ .netscape/ maple ptr/

../ .fvwin2rc95 .term/ mbox sobsrc. tgz

.Xauthority .kermrc Mail/ mm/ telnetd*

.bash-history .ncftp/ fd/ mount tmp/

例4：用长格式列出某个目录下所有的文件（包括隐藏文件）。

$ ls -laF /home/xu

total 584

drwxr-xr-x 12 root root 1024 Nov 02 22:07 ./

drwxr-xr-x 15 bbs bbs 1024 Jul 29 07:08 ../

-rw-r--r-- 1 yu users 4343 Jul 29 22:20 .bash-history

drwx------ 2 yu users 1024 May 17 06:36 .elm/

--rw------ 1 root root 4628 Jun 2 11:34 mbox

lrwxrwxrwx 1 root root 14 Jul 29 03:08 mount->/mnt

drwxrwxr-x 4 root root 1024 Jul 23 03:43 ptr/

-rw--r--r- 1 root root 483997 Jul 15 17:31 sobsrc. tgz

-rwxr-xr-x 1 root users 60177 Jun 8 01:29 telnetd*

drwxr-xr-x 2 root root 1024 Mar 6 22:32 tmp/

例5：用长格式列出某个目录下所有的文件包括隐藏文件和它们的i节点号。并把文 件属主和属组以UID号和GID号的形式显示。

$ ls -lainF /home/xu total 584 399672

drwxr-xr-x 12 0 0 1024 Nov 30 22:07 ./ 333907

drwxr-xr-x 15 9999 99 1024 Jul 29 07:08 ../ 399801

-rw------- 1 0 0 0 Jun 2 2:09 .Xauthority 399679

-rw-r--r-- 1 505 100 4343 Jul 29 22:20 .bash-hist0ry 301763

drwxr----- 2 505 100 1024 May 17 06:36 .elm/ ……

2005-1-20 上一页 首页 下一页

10.Linux改变文件或目录的访问权限命令

我们一起在风雨中前行

Linux改变文件或目录的访问权限命令

Linux系统中的每个文件和目录都有访问许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。

文件或目录的访问权限分为只读，只写和可执行三种。以文件为例，只读权限表示只允许读其内容，而禁止对其做任何的更改操作。可执行权限表示允许将该文件作为一个程序执行。文件被创建时，文件所有者自动拥有对该文件的读、写和可执行权限，以便于对文件的阅读和修改。用户也可根据需要把访问权限设置为需要的任何组合。

有三种不同类型的用户可对文件或目录进行访问：文件所有者，同组用户、其他用户。所有者一般是文件的创建者。所有者可以允许同组用户有权访问文件，还可以将文件的访问权限赋予系统中的其他用户。在这种情况下，系统中每一位用户都能访问该用户拥有的文件或目录。

每一文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写和执行权限；与属主同组的用户的读、写和执行权限；系统中其他用户的读、写和执行权限。当用ls
-l命令显示文件或目录的详细信息时，最左边的一列为文件的访问权限。例如：

$ ls -l sobsrc. tgz

-rw-r--r-- 1 root root 483997 Jul 15 17:31 sobsrc. tgz

横线代表空许可。r代表只读，w代表写，x代表可执行。注意这里共有10个位置。第一个字符指定了文件类型。在通常意义上，一个目录也是一个文件。如果第一个字符是横线，表示是一个非目录的文件。如果是d，表示是一个目录。

例如：

- rw- r-- r--

普通文件 文件主 组用户 其他用户

是文件sobsrc.tgz 的访问权限，表示sobsrc.tgz是一个普通文件；sobsrc.tgz的属主有读写权限；与sobsrc.tgz属主同组的用户只有读权限；其他用户也只有读权限。

确定了一个文件的访问权限后，用户可以利用Linux系统提供的chmod命令来重新设定不同的访问权限。也可以利用chown命令来更改某个文件或目录的所有者。利用chgrp命令来更改某个文件或目录的用户组。

下面分别对这些命令加以介绍。

chmod 命令

功能：chmod命令是非常重要的，用于改变文件或目录的访问权限.用户用它控制文件或目录的访问权限.

语法：该命令有两种用法。一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。

1. 文字设定法

chmod [who] [+ | - | =] [mode] 文件名?

参数：

操作对象who可是下述字母中的任一个或者它们的组合：

u 表示“用户（user）”，即文件或目录的所有者。

g 表示“同组（group）用户”，即与文件属主有相同组ID的所有用户。

o 表示“其他（others）用户”。

a 表示“所有（all）用户”。它是系统默认值。

操作符号可以是：

+ 添加某个权限。

- 取消某个权限。

= 赋予给定权限并取消其他所有权限（如果有的话）。

设置mode所表示的权限可用下述字母的任意组合：

r 可读。

w 可写。

x 可执行。

X 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加x 属性。

s 在文件执行时把进程的属主或组ID置为该文件的文件属主。方式“u＋s”设置文件的用户ID位，“g＋s”设置组ID位。

t 保存程序的文本到交换设备上。

u 与文件属主拥有一样的权限。

g 与和文件属主同组的用户拥有一样的权限。

o 与其他用户拥有一样的权限。

文件名：以空格分开的要改变权限的文件列表，支持通配符。

在一个命令行中可给出多个权限方式，其间用逗号隔开。例如：chmod g+r，o+r example

使同组和其他用户对文件example 有读权限。

2. 数字设定法

我们必须首先了解用数字表示的属性的含义：0表示没有权限，1表示可执行权限，2表示可写权限，4表示可读权限，然后将其相加。所以数字属性的格式应为3个从0到7的八进制数，其顺序是（u）（g）（o）。

例如，如果想让某个文件的属主有"读/写"二种权限，需要把4（可读）+2（可写）＝6（读/写).

数字设定法的一般形式为：

chmod [mode] 文件名?

例子：

（1）文字设定法：

例1：$ chmod a+x sort

即设定文件sort的属性为：

文件属主（u） 增加执行权限

与文件属主同组用户（g） 增加执行权限

其他用户（o） 增加执行权限

例2：$ chmod ug+w，o-x text

即设定文件text的属性为：

文件属主（u） 增加写权限

与文件属主同组用户（g） 增加写权限

其他用户（o） 删除执行权限

例3：$ chmod u+s a.out

假设执行chmod后a.out的权限为（可以用ls – l a.out命令来看）：

–rws--x--x 1 inin users 7192 Nov 4 14:22 a.out

并且这个执行文件要用到一个文本文件shiyan1.c，其文件存取权限为“–rw-------”，即该文件只有其属主具有读写权限。

当其他用户执行a.out这个程序时，他的身份因这个程序暂时变成inin（由于chmod命令中使用了s选项），所以他就能够读取shiyan1.c这个文件（虽然这个文件被设定为其他人不具备任何权限），这就是s的功能。

因此，在整个系统中特别是root本身，最好不要过多的设置这种类型的文件（除非必要）这样可以保障系统的安全，避免因为某些程序的bug而使系统遭到入侵。

例4：$ chmod a–x mm.txt

$ chmod –x mm.txt

$ chmod ugo–x mm.txt

以上这三个命令都是将文件mm.txt的执行权限删除，它设定的对象为所有使用者。

（2）数字设定法：

例1： $ chmod 644 mm.txt

$ ls –l

即设定文件mm.txt的属性为：

-rw-r--r-- 1 inin users 1155 Nov 5 11:22 mm.txt

文件属主（u）inin 拥有读、写权限

与文件属主同组人用户（g） 拥有读权限

其他人（o） 拥有读权限

例2： $ chmod 750 wch.txt

$ ls –l

-rwxr-x--- 1 inin users 44137 Nov 12 9:22 wchtxt

即设定wchtxt这个文件的属性为：

文件主本人（u）inin 可读/可写/可执行权

与文件主同组人（g） 可读/可执行权

其他人（o） 没有任何权限

chgrp命令

功能：改变文件或目录所属的组。

语法：chgrp [选项] group filename?

该命令改变指定指定文件所属的用户组。其中group可以是用户组ID，也可以是/etc/group文件中用户组的组名。文件名是以空格分开的要改变属组的文件列表，支持通配符。如果用户不是该文件的属主或超级用户，则不能改变该文件的组。

参数：

- R 递归式地改变指定目录及其下的所有子目录和文件的属组。

例1：$ chgrp - R book /opt/local /book

改变/opt/local /book/及其子目录下的所有文件的属组为book。

chown 命令

功能：更改某个文件或目录的属主和属组。这个命令也很常用。例如root用户把自己的一个文件拷贝给用户xu，为了让用户xu能够存取这个文件，root用户应该把这个文件的属主设为xu，否则，用户xu无法存取这个文件。

语法：chown [选项] 用户或组 文件

说明：chown将指定文件的拥有者改为指定的用户或组。用户可以是用户名或用户ID。组可以是组名或组ID。文件是以空格分开的要改变权限的文件列表，支持通配符。

参数：

- R 递归式地改变指定目录及其下的所有子目录和文件的拥有者。

- v 显示chown命令所做的工作。

例1：把文件shiyan.c的所有者改为wang。

$ chown wang shiyan.c

例2：把目录/his及其下的所有文件和子目录的属主改成wang，属组改成users。

$ chown - R wang.users /his

http://www.linuxsir.com

11.Linux与用户有关的命令

我们一起在风雨中前行

Linux与用户有关的命令

passwd命令

出于系统安全考虑，Linux系统中的每一个用户除了有其用户名外，还有其对应的用户口令。因此使用useradd命令增加时，还需使用passwd命令为每一位新增加的用户设置口令；用户以后还可以随时用passwd命令改变自己的口令。

该命令的一般格式为： passwd [用户名] 其中用户名为需要修改口令的用户名。只有超级用户可以使用“passwd 用户名”修改其他用户的口令，普通用户只能用不带参数的passwd命令修改自己的口令。

该命令的使用方法如下：

输入

passwd< Enter>；

在（current） UNIX passwd:下输入当前的口令

在new password:提示下输入新的口令（在屏幕上看不到这个口令）：

系统提示再次输入这个新口令。

输入正确后，这个新口令被加密并放入/etc/shdow文件.选取一个不易被破译的口令是很重要的.

选取口令应遵守如下规则：

口令应该至少有六位（最好是八位）字符；

口令应该是大小写字母、标点符号和数字混杂的。

超级用户修改其他用户（xxq）的口令的过程如下，

passwd xxq

New UNIX password:

Retype new UNIX password:

passwd: all authentication tokens updated successfully

#

su命令

这个命令非常重要。它可以让一个普通用户拥有超级用户或其他用户的权限，也可以让超级用户以普通用户的身份做一些事情。普通用户使用这个命令时必须有超级用户或其他用户的口令。如要离开当前用户的身份，可以打exit。

该命令的一般形式为： su [选项] [?] [使用者帐号]

说明：若没有指定使用者帐号，则系统预设值为超级用户root。

参数：

? c 执行一个命令后就结束。

? 加了这个减号的目的是使环境变量和欲转换的用户相同。

? m 保留环境变量不变。

例1： 变成root用户

$ su ?

password: 【输入超级用户的密码】

例2： 变成xu使用者，并执行一个命令就结束。

$ su -xu ? c “rmdir cat1”

http://www.linuxsir.com

12.Linux系统管理命令

我们一起在风雨中前行

Linux系统管理命令

wall命令

功能：这个命令的功能是对全部已登录的用户发送信息，用户可以先把要发送的信息写好存入一个文件中，然后输入：

wall < 文件名

这样就能对所有的用户发送信息了。

在上面的例子中符号“<”表示输入重定向，有关它的含义和用法请参阅第十章的有关内容。 例如：

wall ‘Thank you!’

Broadcast message from root （tty1） Fri Nov 26 14：15：07 1999…

Thank you!

#

执行以上命令后，用户的屏幕上显示出“Thank you!”信息后，并不出现系统提示符$（#），再次按回车键后，屏幕出现系统提示符。

write命令

功能：write命令的功能是向系统中某一个用户发送信息。

格式：write 用户帐号 [终端名称]

例如： $ write xxq hello

此时系统进入发送信息状态，用户可以输入要发送的信息，输入完毕，希望退出发送状态时，按组合键< Ctrl+c>即可。

上述命令执行的结果是，用户xxq的屏幕上会显示：

message from test@test.tlc.com.cn tty1 at 15:51…

hello

EOF

mesg指令

mesg命令设定是否允许其他用户用write命令给自己发送信息。如果允许别人给自己发送信息，输入命令：

mesg y

否则，输入：

mesg n

对于超级用户，系统的默认值为 n；而对于一般用户系统的默认值为y。 如果mesg后不带任何参数，则显示当前的状态是y还是n，如：

$ mesg is

y

或：

mesg is

n

sync命令

功能：sync命令是在关闭Linux系统时使用的。 用户需要注意的是，不能用简单的关闭电源的方法关闭系统，因为Linux象其他Unix系统一样，在内存中缓存了许多数据，在关闭系统时需要进行内存数据与硬盘数据的同步校验，保证硬盘数据在关闭系统时是最新的，只有这样才能确保数据不会丢失。一般正常的关闭系统的过程是自动进行这些工作的，在系统运行过程中也会定时做这些工作，不需要用户干预。

sync命令是强制把内存中的数据写回硬盘，以免数据的丢失。用户可以在需要的时候使用此命令。

格式： sync

shutdown命令

功能：shutdown 命令可以安全地关闭或重启Linux系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。精确时间的格式是hh:mm，表示小时和分钟；时间段由“+”和分钟数表示。系统执行该命令后，会自动进行数据同步的工作。

格式：shutdown [选项] [时间] [警告信息]

参数：

- k 并不真正关机，而只是发出警告信息给所有用户。

- r 关机后立即重新启动。

- h 关机后不重新启动。

- f 快速关机，重启动时跳过fsck。

- n 快速关机，不经过init程序。

- c 取消一个已经运行的shutdown。

需要特别说明的是，该命令只能由超级用户使用。

例1：系统在十分钟后关机，并且马上重新启动。

shutdown – r +10

例2：系统马上关机，并且不重新启动。

shutdown – h now

free命令

功能：free命令的功能是查看当前系统内存的使用情况，它显示系统中剩余及已用的物理内存和交换内存，以及共享内存和被核心使用的缓冲区。

格式： free [-b | -k | -m]

参数：

-b 以字节为单位显示。

-k 以K字节为单位显示。

-m 以兆字节为单位显示。

例：

$ free

total used free shared buffers cached

Mem: 63076 32020 31056 8204 16360 6048

-/+ buffers/cache: 9612 53464

Swap: 64476 2240 62236

uptime命令

功能：uptime命令显示系统已经运行了多长时间，它依次显示下列信息：现在时间、系统已经运行了多长时间、目前有多少登录用户、系统在过去的1分钟、5分钟和15分钟内的平均负载。

格式： uptime

例：

uptime

4:43pm up 1 day, 5:51, 2 user, load average: 0.01, 0.01, 0.00

2005-1-20 上一页 首页 下一页

13.Linux的常用网络命令

我们一起在风雨中前行

Linux 的常用网络命令

计算机网络的主要优点是能够实现资源和信息的共享，并且用户可以远程访问信息。Linux提供了一组强有力的网络命令来为用户服务，这些工具能够帮助用户登录到远程计算机上、传输文件和执行远程命令等。
本章介绍下列几个常用的有关网络操作的命令：
ftp 传输文件
telnet 登录到远程计算机上
r - 使用各种远程命令
netstat 查看网络的状况
nslookup 查询域名和IP地址的对应
finger 查询某个使用者的信息
ping 查询某个机器是否在工作
使用ftp命令进行远程文件传输
ftp命令是标准的文件传输协议的用户接口。ftp是在TCP/IP网络上的计算机之间传输文件的简单有效的方法。它允许用户传输ASCII文件和二进制文件。
在ftp会话过程中，用户可以通过使用ftp客户程序连接到另一台计算机上。从此，用户可以在目录中上下移动、列出目录内容、把文件从远程机拷贝到本地机上、把文件从本地机传输到远程系统中。

需要注意的是，如果用户没有那个文件的存取权限，就不能从远程系统中获得文件或向远程系统传输文件。 为了使用ftp来传输文件，用户必须知道远程计算机上的合法用户名和口令。这个用户名/口令的组合用来确认ftp
会话，并用来确定用户对要传输的文件可以进行什么样的访问。另外，用户显然需要知道对其进行ftp会话的计算机的名字或IP地址。
Ftp命令的功能是在本地机和远程机之间传送文件。该命令的一般格式如下：
$ ftp 主机名/IP

其中“主机名/IP”是所要连接的远程机的主机名或IP地址。在命令行中，主机名属于选项，如果指定主机名，ftp将试图与远程机的ftp服务程序进行连接；如果没有指定主机名，ftp将给出提示符，等待用户输入命令：

$ ftp ftp > 此时在ftp>提示符后面输入open命令加主机名或IP地址，将试图连接指定的主机。 不管使用哪一种方法，如果连接成功，需要在远程机上登录。用户如果在远程机上有帐号，就可以通过ftp使用这一帐号并需要提供口令。

在远程机上的用户帐号的读写权限决定该用户在远程机上能下载什么文件和将上载文件放到哪个目录中。 如果没有远程机的专用登录帐号，许多ftp站点设有可以使用的特殊帐号。这个帐号的登录名为anonymous（也称为匿名ftp），当使用这一帐号时，要求输入email地址作为口令。
如果远程系统提供匿名ftp服务，用户使用这项服务可以登录到特殊的，供公开使用的目录。
一般专门提供两个目录：pub目录和incoming目录。pub目录包含该站点供公众使用的所有文件，incoming目录存放上载到该站点的文件。
一旦用户使用ftp在远程站点上登录成功，将得到“ftp>”提示符。现在可以自由使用ftp提供的命令，可以用 help命令取得可供使用的命令清单，也可以在
help命令后面指定具体的命令名称，获得这条命令的说明。
最常用的命令有：

ls 列出远程机的当前目录
cd 在远程机上改变工作目录
lcd 在本地机上改变工作目录
ascii 设置文件传输方式为ASCII模式
binary 设置文件传输方式为二进制模式
close终止当前的ftp会话
hash 每次传输完数据缓冲区中的数据后就显示一个#号
get（mget） 从远程机传送指定文件到本地机
put（mput） 从本地机传送指定文件到远程机
open 连接远程ftp站点
quit断开与远程机的连接并退出ftp
? 显示本地帮助信息
! 转到Shell中

下面简单将ftp常用命令作一简介。
启动ftp会话 open命令用于打开一个与远程主机的会话。该命令的一般格式是： open 主机名/IP 如果在ftp 会话期间要与一个以上的站点连接，通常只用不带参数的ftp命令。如果在会话期间只想与一台计算机连接，那么在命令行上指定远程主机名或IP地址作为ftp命令的参数。
终止ftp会话 close、disconnect、quit和bye命令用于终止与远程机的会话。close和disronnect命令关闭与远程机的连接，但是使用户留在本地计算机的ftp程序中。quit和bye命令都关闭用户与远程机的连接，然后退出用户机上的ftp
程序。 改变目录 “cd [目录]”命令用于在ftp会话期间改变远程机上的目录，lcd命令改变本地目录，使用户能指定查找或放置本地文件的位置。
远程目录列表 ls命令列出远程目录的内容，就像使用一个交互shell中的ls命令一样。ls命令的一般格式是： ls [目录] [本地文件]
如果指定了目录作为参数，那么ls就列出该目录的内容。如果给出一个本地文件的名字，那么这个目录列表被放入本地机上您指定的这个文件中。 从远程系统获取文件
get和mget命令用于从远程机上获取文件。get命令的一般格式为： get 文件名 您还可以给出本地文件名，这个文件名是这个要获取的文件在您的本地机上创建时的文件名。如果您不给出一个本地文件名，那么就使用远程文件原来的名字。
mget命令一次获取多个远程文件。mget命令的一般格式为： mget 文件名列表 使用用空格分隔的或带通配符的文件名列表来指定要获取的文件，对其中的每个文件都要求用户确认是否传送。
向远程系统发送文件 put和mput命令用于向远程机发送文件。Put命令的一般格式为： put 文件名 mput命令一次发送多个本地文件，mput命令的一般格式为：

mput 文件名列表 使用用空格分隔的或带通配符的文件名列表来指定要发送的文件。对其中的每个文件都要求用户确认是否发送。 改变文件传输模式

默认情况下，ftp按ASCII模式传输文件，用户也可以指定其他模式。ascii和brinary命令的功能是设置传输的模式。用ASCII模式传输文件对纯文本是非常好的，但为避免对二进制文件的破坏，用户可以以二进制模式传输文件。

检查传输状态 传输大型文件时，可能会发现让ftp提供关于传输情况的反馈信息是非常有用的。hash命令使ftp在每次传输完数据缓冲区中的数据后，就在屏幕上打印一个#字符。本命令在发送和接收文件时都可以使用。

ftp中的本地命令 当您使用ftp时，字符“!”用于向本地机上的命令shell传送一个命令。如果用户处在ftp会话中，需要shell做某些事，就很有用。例如用户要建立一个目录来保存接收到的文件。如果输入! mkdir

new_dir，那么Linux就在用户当前的本地目录中创建一个名为new_dir 的目录。

从远程机grunthos下载二进制数据文件的典型对话过程如下：

$ ftp grunthos Connected to grunthos 220 grunthos ftp server Name （grunthos:pc）: anonymous 331 Guest login ok, send your complete e-mail address as password.
Password: 230 Guest login ok, access restrictions apply. Remote system type is UNIX. ftp > cd pub 250 CWD command successful. ftp > ls
200 PORT command successful. 150 opening ASCII mode data connection for /bin/1s. total 114 rog1 rog2 226 Transfer complete . ftp > binary 200 type set to I. ftp > hash Hash mark printing on （1024 bytes/hash mark）.
ftp > get rog1 200 PORT command successful. 150 opening BINARY mode data connection for rogl （14684 bytes）. # # # # # # # # # # # # # # 226 Transfer complete. 14684 bytes received in 0.0473 secs （3e + 02 Kbytes/sec）
ftp > quit 221 Goodbye.

使用telnet命令访问远程计算机

用户使用telnet命令进行远程登录。该命令允许用户使用telnet协议在远程计算机之间进行通信，用户可以通过网络在远程计算机上登录，就像登录到本地机上执行命令一样。

为了通过telnet登录到远程计算机上，必须知道远程机上的合法用户名和口令。虽然有些系统确实为远程用户提供登录功能，但出于对安全的考虑，要限制来宾的操作权限，因此，这种情况下能使用的功能是很少的。当允许远程用户登录时，系统通常把这些用户放在一个受限制的shell中，以防系统被怀有恶意的或不小心的用户破坏。

用户还可以使用telnet从远程站点登录到自己的计算机上，检查电子邮件、编辑文件和运行程序，就像在本地登录一样。

但是，用户只能使用基于终端的环境而不是X Wndows环境，telnet只为普通终端提供终端仿真，而不支持 X Wndow等图形环境。 telnet命令的一般形式为：

telnet 主机名/IP 其中“主机名/IP”是要连接的远程机的主机名或IP地址。如果这一命令执行成功，将从远程机上得到login：提示符。

使用telnet命令登录的过程如下： $ telnet 主机名/IP 启动telnet会话。 一旦telnet成功地连接到远程系统上，就显示登录信息并提示用户输人用户名和口令。如果用户名和口令输入正确，就能成功登录并在远程系统上工作。

在telnet提示符后面可以输入很多命令，用来控制telnet会话过程，在telnet联机帮助手册中对这些命令有详细的说明。

下面是一台Linux计算机上的telnet会话举例：

$ telnet server. somewhere. com Trying 127.0.0.1… Connected to serve. somewhere. com. Escape character is \'?]\'. “TurboLinux release 4. 0 （Colgate）
kernel 2.0.18 on an I486 login: bubba password: Last login:Mon Nov 15 20:50:43 for localhost Linux 2. 0.6. （Posix）. server: ~$ server: ~$

logout Connection closed by foreign host $

用户结束了远程会话后，一定要确保使用logout命令退出远程系统。然后telnet报告远程会话被关闭，并返回到用户的本地机的Shell提示符下。

r-系列命令 除ftp和telnet以外，还可以使用r-系列命令访问远程计算机和在网络上交换文件。 使用r-系列命令需要特别注意，因为如果用户不小心，就会造成严重的安全漏洞。用户发出一个r-系列命令后，远程系统检查名为/etc/hosts.equiv的文件，以查看用户的主机是否列在这个文件中。如果它没有找到用户的主机，就检查远程机上同名用户的主目录中名为.rhosts的文件，看是否包括该用户的主机。如果该用户的主机包括在这两个文件中的任何一个之中，该用户执行r-系列命令就不用提供口令。

虽然用户每次访问远程机时不用键入口令可能是非常方便的，但是它也可能会带来严重的安全问题。我们建议用户在建立/etc/hosts.equiv和.rhosts文件之前，仔细考虑r-命令隐含的安全问题。

rlogin命令

功能：rlogin 是“remote login”（远程登录）的缩写。该命令与telnet命令很相似，允许用户启动远程系统上的交互命令会话。

格式： rlogin [-8EKLdx] [-e char] [-k
realm] [- l username] host

参数：

-8 此选项始终允许8位输入数据通道。该选项允许发送格式化的ANSI字符和其他的特殊代码。如果不用这个选项，除非远端的终止和启动字符不是或，否则就去掉奇偶校验位。

-E 停止把任何字符当作转义字符。当和-8选项一起使用时，它提供一个完全的透明连接。

-K 关闭所有的Kerberos确认。只有与使用Kerberos 确认协议的主机连接时才使用这个选项。

-L 允许rlogin会话在litout模式中运行。要了解更多信息，请查阅tty联机帮助。

-d 打开与远程主机进行通信的TCP sockets的socket调试。要了解更多信息，请查阅setsockopt的联机帮助。

-e 为rlogin会话设置转义字符，默认的转义字符是“~”，用户可以指定一个文字字符或一个\\nnn形式的八进制数。

-k 请求rlogin获得在指定区域内的远程主机的Kerberos许可，而不是获得由krb_realmofhost（3）确定的远程主机区域内的远程主机的Kerberos
许可。

-x 为所有通过rlogin会话传送的数据打开DES加密。这会影响响应时间和CPU利用率，但是可以提高安全性。

rsh命令

功能：rsh是“remote shell”（远程 shell）的缩写。 该命令在指定的远程主机上启动一个shell并执行用户在rsh命令行中指定的命令。如果用户没有给出要执行的命令，rsh就用rlogin命令使用户登录到远程机上。

格式：rsh [-Kdnx] [-k realm] [-l username]
host [command]

command可以是从shell提示符下键人的任何Linux命令。

参数：

-K 关闭所有的Kerbero确认。该选项只在与使用Kerbero确认的主机连接时才使用。

-d 打开与远程主机进行通信的TCP sockets的socket调试。要了解更多的信息，请查阅setsockopt的联机帮助。

-k 请求rsh获得在指定区域内的远程主机的Kerberos许可，而不是获得由krb_relmofhost（3）确定

的远程主机区域内的远程主机的Kerberos许可。

-l 缺省情况下，远程用户名与本地用户名相同。本选项允许指定远程用户名，如果指定了远程用户名，则使用Kerberos 确认，与在rlogin命令中一样。

-n 重定向来自特殊设备/dev/null的输入。

-x 为传送的所有数据打开DES加密。这会影响响应时间和CPU利用率，但是可以提高安全性。Linux把标准输入放入rsh命令中,并把它拷贝到要远程执行的命令的标准输入中。它把远程命令的标准输出拷贝到rsh的标准输出中。它还把远程标准错误拷贝到本地标准错误文件中。任何退出、中止和中断信号都被送到远程命令中。当远程命令终止了，rsh也就终止了。

rcp命令

功能：rcp代表“remote file copy”（远程文件拷贝）。该命令用于在计算机之间拷贝文件。

rcp命令有两种格式。第一种格式用于文件到文件的拷贝；第二种格式用于把文件或目录拷贝到另一个目录中。

格式：

rcp [-px] [-k realm] file1 file2 rcp [-px] [-r] [-k realm] file directory 每个文件或目录参数既可以是远程文件名也可以是本地文件名。远程文件名具有如下形式：rname@rhost：path，其中rname是远程用户名，rhost是远程计算机名，path是这个文件的路径。

参数：

-r 递归地把源目录中的所有内容拷贝到目的目录中。要使用这个选项，目的必须是一个目录。

-p 试图保留源文件的修改时间和模式，忽略umask。

-k 请求rcp获得在指定区域内的远程主机的Kerberos 许可，而不是获得由krb_relmofhost（3）确定的远程主机区域内的远程主机的Kerberos许可。

-x 为传送的所有数据打开DES加密。这会影响响应时间和CPU利用率，但是可以提高安全性。 如果在文件名中指定的路径不是完整的路径名，那么这个路径被解释为相对远程机上同名用户的主目录。如果没有给出远程用户名，就使用当前用户名。如果远程机上的路径包含特殊shell字符，需要用反斜线（\\）、双引号（”）或单引号（’）括起来，使所有的shell元字符都能被远程地解释。
需要说明的是，rcp不提示输入口令，它通过rsh命令来执行拷贝。 - Turbolinux 提供稿件

系列教材四

目录

00.使用技巧集

1、处理特殊的文件名

假设Linux系统中有一个文件名叫“-ee”，如果我们想对它进行操作，例如要删除它，按照一般的删除方法在命令行中输入rm -ee命令，界面会提示我们是“无效选项”(invalid option)，原来由于文件名的第一个字符为“-”，Linux把文件名当作选项了，我们可以使用“--”符号来解决这个问题，输入“rm -- -ee”命令便可顺利删除名为“-ee”的文件。如果是其他特殊字符的话可以在特殊字符前加一个“”符号，或者用双引号把整个文件名括起来。

2、直接进行Linux的安装工作

在安装Linux操作系统时，可以利用该系统光盘中的一个名为“loadlin.exe”的软件，将Linux核心直接调入内存，由Linux核心代替当前操作系统来接管计算机，并进入Linux的安装界面。在安装Linux时，我们只要在运行对话框中输入“loadlin E:imagesvmlinuz root=/dev/ram initrd=E:imagesinitrd.img”这个命令就可以直接安装Linux了；其中“E:imagesvmlinuz”为Linux的核心名。

3、消除Xwindows下的死机现象

我们可以用两个常用的方法来消除这种现象：第一，用键盘上的复合键“Ctrl+Alt+Backspace”来关闭当前正在运行的任务；第二，首先按住键盘上的“Ctrl+Alt+F2”复合键，让系统切换到另一个操作台，然后登录到系统，再执行“#ps -ax/grep startx”命令，这将会列出你的Xserver的进程标识，接着在命令行中输入如下命令就能消除Xwindows下的死机现象：#kill -9 PID_Number，最后通过“Alt+F1”复合键返回到原来的平台。

4、快速关闭Linux系统

最新版本的Linux/UNIX系统借鉴了大型机的技术，采用了抗掉电的日志式文件系统，可以自动跟踪保存用户数据，自动同步刷新文件系统，用户完全可以随手关闭电源，从而达到快速关闭系统的目的。

5、巧妙使用“rm”命令

我们可以使用带“-r”参数的“rm”命令来删除一个非空目录，例如我们在命令行中输入“rm -r bbb”这样的命令，表示系统将把bbb目录中包含的所有文件和子目录全部删除掉。

6、巧妙使用“Tab”键

大家知道在Linux字符界面中输入命令时，有时需要输入很多字符，如果经常这样逐个地输入字符，比较麻烦。假设键入的字符足以确定该目录下一个惟一的文件时，我们只需按键盘上的“Tab”键就可以自动补齐该文件名的剩下部分，例如要把目录/ccc下的文件“ddddddd-1.2.3.tar.gz”解包时，当我们在命令行中键入到“tar
xvfz /ccc/d”时，如果该文件是该目录下惟一以“d”打头的文件的话就可以直接按下“Tab”键，这时命令会被自动补齐为：tar xvfz
/ccc/ddddddd-1.2.3.tar.gz ，从而提高了输入效率。

7、多用鼠标拷贝与粘贴来提高操作速度

Linux系统安装后，每次启动到字符界面时都会自动运行一个叫“gpm”的程序，该程序运行后就可以用鼠标来拷贝与粘贴了。具体做法是按住鼠标左键拖动使要拷贝的地方突出显示，这时突出显示的区域已经被拷贝，再按鼠标右键拷贝的内容就会被粘贴在光标所在位置了。如果我们在Xwindow下运行Linux系统，拷贝与粘贴的操作与在Windows
9x系统下一样。

8、快速启动Linux系统

在DOS下，有一种简单快速启动Linux的方法，那就是load Linux。loadlin.exe是DOS下的可执行程序，它可以在纯DOS环境下迅速启动Linux，而且无需重启计算机，通常我们可以在光盘的“/kernels”目录下找到这个程序。如果不知这个程序被放置于安装盘的何处，可以使用“find
-name loadlin*”命令来寻找。找到之后将其复制到DOS分区中，同时还需要复制一份你所使用的Linux内核文件。可以通过Windows直接从光盘复制，也可在Linux环境下使用“mcopy”命令将文件复制到DOS分区；接着再编写一个名为“Linux.bat”的批处理文件，文件内容如下：c:loadlin
c:vmlinuz root=/dev/hda1 ro(我们假设loadlin.exe和vmlinuz这两个内核文件都在c盘根目录下，“root”为Linux的根设备，而且Linux处于硬盘第一分区，所以设备名为“/dev/hda1”，“ro”意为readonly)。以后在DOS下要启动Linux时，运行“Linux.bat”就可以了。

2005-1-20 11:41:12

上一页 首页 下一页

2005-1-20

上一页 首页 下一页

01.使用技巧集一

我们一起在风雨中前行

问：Linux为何无法支持34GB以上的IDE硬盘？

答：这是kernel版本的问题，在核心版本2．2．14之前对于IDE接口的硬盘只能支持到34GB，到核心版本2．2．14以上(包括2．2．14版本)便能支持34GB以上IDE接口的硬盘。

问：Linux中如何检测CPU型号？

答：键入cat /proc/cpuinfo命令即可获知系统配备了几个CPU、CPU的型号、制造商、运行频率、cache的大小、有无fpu等信息。

问：如何让鼠标的按键符合左手使用的方向？

答：其实，在x-window中，我们可以通过对鼠标的设定来达到鼠标按键方向对调的效果。你可以在.xinitrc中加上下面一行内容就可调整鼠标按键方向：

xmodmap -e "pointer = 3 2 1"

问：如何处理才能在开机后就去执行一个程序？

答：请在/etc/rc.d/rc.local中写入你要执行的程序。例如一开机便启动MySQL数据库，请在/etc/rc.d/rc.local下键入：

/usr/local/bin/mysql.server start即可。注意，请不要使用：exec /usr/local/bin/mysql.server
start

因为exec会使得后面要执行的程序取代目前的程序，如此一来，放在/usr/local/bin/mysql.server start后面的程序就都不会执行了。

问：如何使得系统在一段时间不操作后能自动退出？

答：当系统管理员 root账户拥有者 在离开计算机时，出于安全考虑，最好能让系统在隔一段时间后能自动退出。为了能做到这一点，你必须为一个叫做"TMOUT"的Linux变量设置指定时间 时间单位是秒 。编辑"/etc/profile"文件，在有"HISTFILESIZE="字样的那一行的后面加上下面一行内容：

TMOUT=3600

加入的这一行代表的含义是1小时 60×60=3600秒 。当你把这行内容放入你的"/etc/profile"文件后，在系统连续一小时不用时，系统会自动通知系统中的所有用户系统将退出。你也可以把该变量设置放在用户的各自的".bashrc"文件中，使得系统能在指定的一段时间不用后能自动退出。

该变量参数被设置在系统中后，你必须先退出系统，然后再以root帐户重新登录后，该项设置才会生效。

问：在Linux下如何更改系统时间？

答：若你要改时间，以root身份执行：

date -s HH MM SS（更改时间，其中HH、MM、SS分别为时、分、秒）

clock -w(写入Cmos)

如此即可，若你是时区设错的话，则可到#timeconfig中重新设定。

02.使用技巧集二

我们一起在风雨中前行

1、处理特殊的文件名

假设Linux系统中有一个文件名叫“-ee”，如果我们想对它进行操作，例如要删除它，按照一般的删除方法在命令行中输入rm
-ee命令，界面会提示我们是“无效选项”(invalid option)，原来由于文件名的第一个字符为“-”，Linux把文件名当作选项了，我们可以使用“--”符号来解决这个问题，输入“rm
-- -ee”命令便可顺利删除名为“-ee”的文件。如果是其他特殊字符的话可以在特殊字符前加一个“”符号，或者用双引号把整个文件名括起来。

2、直接进行Linux的安装工作

在安装Linux操作系统时，可以利用该系统光盘中的一个名为“loadlin.exe”的软件，将Linux核心直接调入内存，由Linux核心代替当前操作系统来接管计算机，并进入Linux的安装界面。在安装Linux时，我们只要在运行对话框中输入“loadlin
E:imagesvmlinuz root=/dev/ram initrd=E:imagesinitrd.img”这个命令就可以直接安装Linux了；其中“E:imagesvmlinuz”为Linux的核心名。

3、消除Xwindows下的死机现象

我们可以用两个常用的方法来消除这种现象：第一，用键盘上的复合键“Ctrl+Alt+Backspace”来关闭当前正在运行的任务；第二，首先按住键盘上的“Ctrl+Alt+F2”复合键，让系统切换到另一个操作台，然后登录到系统，再执行“#ps
-ax/grep startx”命令，这将会列出你的Xserver的进程标识，接着在命令行中输入如下命令就能消除Xwindows下的死机现象：#kill
-9 PID_Number，最后通过“Alt+F1”复合键返回到原来的平台。

4、快速关闭Linux系统

最新版本的Linux/UNIX系统借鉴了大型机的技术，采用了抗掉电的日志式文件系统，可以自动跟踪保存用户数据，自动同步刷新文件系统，用户完全可以随手关闭电源，从而达到快速关闭系统的目的。

5、巧妙使用“rm”命令

我们可以使用带“-r”参数的“rm”命令来删除一个非空目录，例如我们在命令行中输入“rm
-r bbb”这样的命令，表示系统将把bbb目录中包含的所有文件和子目录全部删除掉。

6、巧妙使用“Tab”键

大家知道在Linux字符界面中输入命令时，有时需要输入很多字符，如果经常这样逐个地输入字符，比较麻烦。假设键入的字符足以确定该目录下一个惟一的文件时，我们只需按键盘上的“Tab”键就可以自动补齐该文件名的剩下部分，例如要把目录/ccc下的文件“ddddddd-1.2.3.tar.gz”解包时，当我们在命令行中键入到“tar
xvfz /ccc/d”时，如果该文件是该目录下惟一以“d”打头的文件的话就可以直接按下“Tab”键，这时命令会被自动补齐为：tar xvfz
/ccc/ddddddd-1.2.3.tar.gz ，从而提高了输入效率。

7、多用鼠标拷贝与粘贴来提高操作速度

Linux系统安装后，每次启动到字符界面时都会自动运行一个叫“gpm”的程序，该程序运行后就可以用鼠标来拷贝与粘贴了。具体做法是按住鼠标左键拖动使要拷贝的地方突出显示，这时突出显示的区域已经被拷贝，再按鼠标右键拷贝的内容就会被粘贴在光标所在位置了。如果我们在Xwindow下运行Linux系统，拷贝与粘贴的操作与在Windows
9x系统下一样。

8、快速启动Linux系统

在DOS下，有一种简单快速启动Linux的方法，那就是load
Linux。loadlin.exe是DOS下的可执行程序，它可以在纯DOS环境下迅速启动Linux，而且无需重启计算机，通常我们可以在光盘的“/kernels”目录下找到这个程序。如果不知这个程序被放置于安装盘的何

处，可以使用“find
-name loadlin*”命令来寻找。找到之后将其复制到DOS分区中，同时还需要复制一份你所使用的Linux内核文件。可以通过Windows直接从光盘复制，也可在Linux环境下使用“mcopy”命令将文件复制到DOS分区；接着再编写一个名为“Linux.bat”的批处理文件，文件内容如下：c:loadlin
c:vmlinuz root=/dev/hda1 ro(我们假设loadlin.exe和vmlinuz这两个内核文件都在c盘根目录下，“root”为Linux的根设备，而且Linux处于硬盘第一分区，所以设备名为“/dev/hda1”，“ro”意为readonly)。以后在DOS下要启动Linux时，运行“Linux.bat”就可以了。

03.使用技巧集三

我们一起在风雨中前行

对于那些用惯了Windows的朋友，初用Linux时可能有处处不方便的感觉，以下是我使用Linux的一些经验技巧，相信对初学者会有所帮助的。

1.使用虚拟控制台
登录后按Alt＋F2键这时又可以看到“login:”提示符，这个就是第二个虚拟控制台。一般新安装的Linux有四个虚拟控制台，可以用Alt＋F1～Alt＋F4来访问。虚拟控制台最有用的时候是当一个程序出错锁住输入时可以切换到其他虚拟控制台登录进入后杀掉这个出错的进程。

2.拷贝与粘贴
适用于字符界面。Linux每次启动时都会自动运行一个叫gpm的程序，该程序运行后就可以用鼠标来拷贝与粘贴了。具体做法是按住鼠标左键拖动使要拷贝的地方反白，这时反白的区域已经被拷贝，再按鼠标右键拷贝的内容就会被粘贴在光标所在位置了。

如果系统启动时，未运行gpm，您可以在进入字符界面后再进行设置：＃ntsysv。将会出现一对话框，先用键盘上的下键头移到gpm选项，然后按空格键将其选中（选中项前有一星号），最后重启，重新登陆即可。

3.快速进入某些目录
键入cd ～可进入用户的home目录；键入cd －可进入上一个进入的目录。
4.软驱的使用
（1）如果是Linux的ext2文件系统用如下命令：＃mount －t ext2 /dev/fd0/mnt/floppy；若是dos格式的软盘则用命令：＃mount
－t msdos/dev/fd0/mnt/floppy。然后就可以在/mnt/floppy里访问软盘的内容了；
（2）注意在取出软盘之前要先卸掉软盘上的文件系统：＃umount /mnt/floppy，否则会导致信息丢失；

（3）在软盘上建立文件系统可用如下命令：＃mke2fs /dev/fd0H1440；
（4）格式化1.44M的软盘：＃fdformat /fd0H1440。

5.使用光驱

（1）使用光驱前，要先mount一下：＃mount /dev/cdrom /mnt/cdrom，然后您就可以进入/mnt/cdrom目录下读取光盘内容了；

（2）当您想退出光盘时，须使用umout命令，否则光驱就会一直处于死锁状态：＃umount /mnt/cdrom。还有一方法可直接退出光盘：＃eject。

6.显示文件的类型

用命令file可以使你知道某个文件究竟是ELF格式的可执行文件，还是shell script文件或是其他的什么格式，例如：＃file startx。

7.命令文件的路径

当你键入一些命令如find、reboot、mount、startx时是否想过这些文件究竟放在哪个目录里？用命令whereis可以实现。如：＃whereis
startx。

8.查找文件

find命令要加一些参数才能查到文件。如：在/usr里查找文件XF86Setup：＃find /usr －name XF86Setup －print。

9.一次解开.tar.gz文件

解开.tar.gz文件一般要先用gunzip解压再用tar解包，在Linux下可以一次完成。如：＃tar zxpvf sample.tar.gz。

10.Linux里的Norton Commander

在提示符下键入命令mc后你就会看到一个与Norton Commander很相似的界面，实际上功能也很相似甚至更强大，比如可以直接对.tar.gz压缩包里的文件进行操作（有点像ZipMagic）。

11.启动后直接进入X－windows

编辑/etc/inittab文件，把id:3:initdefautl改为id:4:initdefautl。不过搜索路径可能会有些问题，关机时你得直接进入/sbin里去执行shutdown。

12.后台运行X程序

执行一个X程序时别忘了在后面加一个＆号，如＃netscape＆。否则的话在你退出执行的程序之前再无法在那个终端窗口里输入命令了。

13. 强行退出X－windows

有时候在X里由于程序出错鼠标键盘都不起作用，这时候不用着急，因为在Linux下几乎不会像在Win95里那样恶性死机，你只须键入Ctrl＋Alt＋BackSpace键就可以回到字符界面下了。

14.显卡的配置（命令：Xconfigurator，注意：X是大写！）

此命令是一图型界面的对话框，您可以根据实际情况选择配置参数。

15.声卡的配置（命令：sndconfig）

此命令也是一图型界面的对话框，您可以根据实际情况选择配置参数，不过Linux兼容的声卡种类比较少。

16.Linux的在线帮助（命令：man 命令名）

Linux提供了几千页的在线帮助文档，内容几乎覆盖了Linux的方方面面，对于初学者很有帮助！例如：

#man mount。
相信以上几点介绍，能给读者带来帮助和启迪！（山东 尹植平）

04.使用技巧集四

我们一起在风雨中前行

作为桌面操作系统，Linux的人机界面可真是不敢恭维，但是，作为网络操作系统，其易用性(对于NOS而言)和高性能恐怕是很难有能出其右的。当然，这并不是说它的操作很简单，而是说若能掌握一些技巧，无论是系统设置还是网络管理都能变得轻松自如、随心所欲。

禁止终端模式下的显示器自动节能
大部分PC上安装Linux后，在终端模式下也能实现自动关闭显示器的功能，但是假如我们要在显示器上即时显示一些信息，如：网络流量、包分析等，这个功能就变得非常讨厌了。采用修改CMOS和系统属性都无法解决，解决的办法是用setterm，设置终端的命令，如setterm
--blank，就能关闭自动节能这一特性。
将man帮助文档存为普通txt文件

Man的地位在Linux中可谓至关重要，比Windows里的中看不中用的Help好多了。但是它所显示的信息却不是普通的文本，如果直接将这些文字重定向到一个文本文件，你会发现，在man中高亮显示的文字就变成了两个，而且有不计其数的制表符，这使我们的打印、编辑都变得非常不便。不过使用下面这样一条语句就能得到tcsh命令的普通帮助文本：man
tcsh|col -b >tcsh.txt，再也不会有那些讨厌的重复字符了。
IDE硬盘的优化
Linux可能更加注重数据的安全性而非性能，这对NOS来说是完全必要的，但是打开IDE硬盘的一些有用的优化应该没什么害处吧。看看这个命令：hdparm，它的作用是显示、设置硬盘(包括光驱)的参数。我们可以先用hdparm
-i /dev/hda查看hda这块硬盘的参数设置，再根据情况来修改，如：hdparm – c 1可将硬盘的I/O读写支持由16位改为32位，hdparm
-m 16是开启硬盘的MultiCount。在执行这条命令之前最好先用-i参数确定你的硬盘所能支持的最大MultiCount数。主要优化的就是这两项了，其他的不敢确定也别乱来。可以用hdparm
-t /dev/had在优化前后测试硬盘性能。
允许root从远程登录
Linux的默认安全设置是root不能从远程工作站登录，这使我们对Linux服务器进行远程管理造成了不小的麻烦，/etc/securtty正是控制这一机制的关键文件，在这个配置文件的root小节中加入pts/0、pts/1等你希望root能够登录的终端名就行了。

恢复被覆盖的MBR(主引导扇区)的Linux引导程序LILO

如果LILO安装在MBR，那么先装Linux再安装Windows，LILO就会被覆盖，这时我们将无法从硬盘引导Linux，怎么办？我们可以启动到Windows命令行模式，从Linux光盘上拷贝loadlin.exe和vmlinuz两个文件到Windows分区，然后执行loadlin
vmlinuz root=/dev/had? rw。这条命令的作用是从指定的分区启动Linux，并将根分区(1)设为读写模式，这样就可以修改Linux的大部分设置或者恢复LILO。要注意的是这种情况启动的Linux并不完整，如：MSDOS、VFAT文件系统的支持、网络等都无法使用，不过没关系，只要能这样启动，我们可以利用mkbootdisk来创建Linux启动盘，用这张启动盘启动Linux，就能完全使用Linux了。启动Linux以后，只要将/etc/lilo.conf文件的第一项boot=/dev/hda?改为boot=/dev/hda，然后执行lilo就能恢复从MBR启动Linux。

配置文件一览表 文件位置及名称
作用
/etc/rc.d/rc.sysinit┤
系统启动时的公用服务配置
/etc/rc.d/rc.local
系统启动时的本地服务配置
/etc/bashrc、/etc/profile
bash shell的公用配置
/home/username/.bashrc
各用户私人bash shell配置
/etc/securetty
终端安全设置
/etc/hosts
主机名与ip地址的对应文件
/etc/resolv.conf
dns服务器配置
/etc/squid.conf或/etc/squid/squid.conf
squid代理服务器软件配置
/etc/smb.conf或/etc/samba/smb.conf
基于smb的网上邻居配置
/etc/dhcpd.conf
linux下的dhcp服务器配置文件
/etc/filesystems
系统支持的文件系统列表
/etc/fstab
当前系统中已激活分区的信息
/etc/host.conf
指定系统搜索dns记录的配置文件
/etc/lilo.conf
启动程序lilo的配置文件

Linux系统的大部分控制都是基于配置文件的，了解这些配置文件能更有利于使用Linux。

将Linux加入到Win NT/2000的启动选单中

Linux的启动选单不错，可惜不能显示中文，而且不能选择启动NT还是Win 98，所以我认为启动选单

还是用NT系列的好。其实做起来很简单，安装完所有Windows系列操作系统后再装Linux，并在安装时选择将LILO安装到Linux所在的分区而非MBR，然后利用启动盘启动Linux，将Windows分区mount为/mnt/dos；然后执行dd if=/dev/hda? bs=512 count=1 of=/mnt/dos/ bootsect.lin，这样在Windows分区就有了一个bootsect.lin文件，它记录了Linux分区的启动信息，然后在Windows NT的启动配置文件boot.ini中加入C:BOOTSECT.LIN = "红帽子Linux 7.0"，再启动电脑的时候，我们发现，在Windows NT启动选单中多了“红帽子Linux 7.0”这个选项，选择它就能启动我们的Linux。需要注意的是NT在启动时仍然只识别8.3格式的文件，所以不要把bootsect.lin改为长文件名。

用Squid来控制文件类型的访问

Linux的一大好处就是能够非常方便地对IP网络进行路由，很多Linux系统就是专门用作Internet接入路由器。但是，iptables或ipchain等包过滤软件只能控制用户访问哪些站点，而无法控制用户访问哪些文件。使用Linux下的代理服务器软件Squid可以很容易地做到这一点，Squid.conf文件中的urlpath_regex项就是控制每种注册的MIME文件访问的，例如以下语句：

```
#给出需要控制访问的MIME文件类型(访问控制列表即acl)
acl denymine urlpath_regex .exe .zip
#禁止这类文件的访问
http_access deny denymine
```

这样就禁止了这条语句以后的所有用户对相应站点zip、exe文件的访问，当然你可以加入其他文件类型，如MP3等。再利用iptables之类的ReDirect重定向功能，强迫用户通过Squid代理上网，即所谓“透明代理”，就可完全控制用户对站点的访问了。

让系统使用默认的彩色文件显示

Linux终端下的不同类型文件的彩色显示让我们感觉非常方便，但有时可能不小心丢失这一功能，在配置文件中加入alias ls="ls --color -F -N"就能保证这一功能的正常使用。

利用cron保持某些分区的激活状态

cron相当于Win 98的计划任务，它能根据用户的要求定时执行某些命令。如果你mount了一些NTFS分区，并经常访问这个分区(如Web Server对一些文件的访问)，很有可能因为时间过长而无法正确读写数据，这时可以利用cron的定时访问功能，保证该分区总是激活的。根据个人经验，网络上的NTFS分区最好让cron每15分钟访问一次，否则就有可能出现需要该分区数据时出现mount失败的情况。

注：本文中凡未经说明的都以RedHat Linux为准。

05.使用技巧集五

我们一起在风雨中前行

RPM 有五种基本的操作方式(不包括创建软件包): 安装, 卸载, 升级, 查询,和验证。 下面我们就来逐一的讲解吧。

一、 安装RPM包

RPM 软件包通常具有类似foo-1.0-1.i386.rpm 的文件名。其中包括 软件包的名称(foo)，版本号(1.0)，发行号(1)，
和 硬件平台(i386)。安装一个软件包只需简单的键入以下命令:

$ rpm -ivh foo-1.0-1.i386.rpm

foo ####################################

RPM安装完毕后会打印出软件包的名字(并不一定与文件名相同)， 而后打印一连串的#号以表示安装进度。虽然软件包的安装被设计的尽量简单易行， 但还是可能会发生如下的错误:

1、 软件包已被安装

如果您的软件包已被安装, 将会出现以下信息:

$ rpm -ivh foo-1.0-1.i386.rpm

foo package foo-1.0-1 is already installed

error: foo-1.0-1.i386.rpm cannot be installed

如果您仍旧要安装该软件包，可以在命令行上使用--replacepkgs 选项，RPM将忽略该错误信息强行安装。

2、文件冲突

如果您要安装的软件包中有一个文件已在安装其它软件包时安装，会出现以下错误信息:

rpm -ivh foo-1.0-1.i386.rpm

foo /usr/bin/foo conflicts with file from bar-1.0-1

error: foo-1.0-1.i386.rpm cannot be installed

要想让RPM 忽略该错误信息， 请使用--replacefiles 命令行选项。

3、未解决依赖关系

RPM软件包可能依赖于其它软件包，也就是说要求在安装了特定的软件包之后才能安装该软件包。如果在您安装某个软件包时存在这种未解决的依赖关系。会产生以下信息:

$ rpm -ivh bar-1.0-1.i386.rpm

failed dependencies: foo is needed by bar-1.0-1

您必须安装完所依赖的软件包，才能解决这个问题。如果您想强制安装(这是个坏主意， 因为安装后的软件包未必能正常运行)， 请使用--nodeps 命令行选项。

二、卸载RPM包

卸载软件包就象安装软件包时一样简单:

$ rpm -e foo

注意这里使用软件包的名字name ``foo''， 而不是软件包文件的名字file ``foo-1.0-1.i386.rpm''。

如果其它软件包依赖于您要卸载的软件包，卸载时则会产生错误信息。如:

$ rpm -e foo

removing these packages would break dependencies:foo is needed by bar-1.0-1

若让RPM忽略这个错误继续卸载(这可不是一个好主意，因为依赖于该软件包的程序可能无法运行)，请使用--nodeps 命令行选项。

三、升级RPM包

升级软件包和安装软件包十分类似：.

$ rpm -Uvh foo-2.0-1.i386.rpm

foo ####################################

RPM将自动卸载已安装的老板本的foo 软件包，您不会看到有关信息。事实上您可能总是使用 -U 来安装软件包，因为即便以往未安装过该软件包，也能正常运行。因为RPM

执行智能化的软件包升级，自动处理配置文件，您将会看到如下信息：

saving /etc/foo.conf as /etc/foo.conf.rpmsave

这表示您对配置文件的修改不一定能向上兼容。 因此，RPM 会先备份老文件再安装新文件。您应当尽快解决这两个配置文件的不同之处，以使系统能持续正常运行。

因为升级实际包括软件包的卸载与安装两个过程，所以您可能会碰到由这两个操作引起的错误。另一个你可能碰到的问题是：当您使用旧版本的软件包来升级新版本的软件时，RPM会产生以下错误信息：

$ rpm -Uvh foo-1.0-1.i386.rpm

foo package foo-2.0-1 (which is newer) is already installed

error: foo-1.0-1.i386.rpm cannot be installed

如果你确有需要将该软件包”降级，加入 --oldpackage 命令选项就可以了。

四、 查询已安装的软件包

使用命令rpm -q来查询已安装软件包的数据库。简单的使用命令 rpm -q foo 会打印出foo软件包的包名

，版本号，和发行号：

$ rpm -q foo

foo-2.0-1

除了指定软件包名以外，您还可以使用以下选项来指明要查询哪些软件包的信 息。 这些选项被称之为“软件包指定选项“。

- -a 查询所有已安装的软件包
- -f 将查询包含有文件. 的软件包
- -p 查询软件包文件名为的软件包

还可以指定查询软件包时所显示的信息。它们被称作信息选择选项：

- -i 显示软件包信息，如描述, 发行号, 尺寸, 构建日期, 安装日期, 平台, 以及其它一些各类信息。
- -l 显示软件包中的文件列表。
- -s 显示软件包中所有文件的状态。
- -d 显示被标注为文档的文件列表(man 手册, info 手册, README's, etc)。
- -c 显示被标注为配置文件的文件列表。这些是您要在安装完毕以后加以定制的文件(sendmail.cf, passwd, inittab, etc)。

对于那些要显示文件列表的文件, 您可以增加-v 命令行选项以获得如同 ls -l 格式的输出。

五、 验证软件包

验证软件包是通过比较已安装的文件和软件包中的原始文件信息来进行的。验证主要是比较文件的尺寸, MD5 校验码,

文件权限, 类型, 属主和用户组等。

rpm-V命令用来验证一个软件包。您可以使用任何包选择选项来查询您要验证的软件包。 命令rpm -V

foo 将用来验证foo软件包。又如：

• 验证包含特定文件的软件包：

rpm -Vf /bin/vi

• 验证所有已安装的软件包：

rpm -Va

• 根据一个RPM包来验证：

rpm -Vp foo-1.0-1.i386.rpm

如果您担心你的RPM数据库已被破坏，就可以使用这种方式。

如果一切均校验正常将不会产生任何输出。如果有不一致的地方，就会显示出来。 输出格式是8位长字符串，``c'' 用以指配置文件，接着是文件名.

8位字符的每一个 用以表示文件与RPM数据库中一种属性的比较结果 。``.'' (点) 表示测试通过。.下面的字符表示对RPM软件包进行的某种测试失败：

显示字符 错误源

5 MD5 校验码

S 文件尺寸

L 符号连接

T 文件修改日期

D 设备

U 用户

G 用户组

M 模式e (包括权限和文件类型)

如果有错误信息输出，您应当认真加以考虑，是通过删除还是重新安装来解决出现的问题。

六、教你一招

RPM不仅是安装/卸载程序的工具，它还是系统维护和诊断的一把好手。看过下面几个例子你就会领教它的厉害了。

• 如果您误删了一些文件，但您不能肯定到底都删除了那些文件，怎么办？ 您可以键入：

rpm -Va

rpm会在屏幕上显示出文件删除的情况。若你发现一些文件丢失了或已被损坏，您就可以重新安装或先卸载再安装该软件包。

• 如果您碰到了一个自己不认识的文件，要想查处它属于哪个软件包，您可以输入以下命令

rpm -qf /usr/X11R6/bin/xjewel

输出的结果会是：

xjewel-1.6-1

• 如果发生综合以上两个例子的情况，如文件/usr/bin/paste出了问题。您想知道哪个软件包中包含该文件，您这时可以简单的键入：

rpm -Vf /usr/bin/paste

• 如果您想了解正在使用的程序的详细信息，您可以键入如下命令来获得软件包中关于该程序的文档信息：

rpm -qdf /usr/bin/ispell

输出结果为:

/usr/man/man4/ispell.4

/usr/man/man4/english.4

/usr/man/man1/unsq.1

/usr/man/man1/tryaffix.1

/usr/man/man1/sq.1

/usr/man/man1/munchlist.1

/usr/man/man1/ispell.1
/usr/man/man1/findaffix.1
/usr/man/man1/buildhash.1
/usr/info/ispell.info.gz
/usr/doc/ispell-3.1.18-1/README

- 您发现了一个新的koules RPM，但您不知道它是做什么的，您可以键入如下命令：

rpm -qip koules-1.2-2.i386.rpm

- 现在您想了解koules的 RPM包在系统里安装了哪些文件， 您可以键入：

rpm -qlp koules-1.2-2.i386.rpm输出结果为：
/usr/man/man6/koules.6
/usr/lib/games/kouleslib/start.raw
/usr/lib/games/kouleslib/end.raw
/usr/lib/games/kouleslib/destroy2.raw
/usr/lib/games/kouleslib/destroy1.raw
/usr/lib/games/kouleslib/creator2.raw
/usr/lib/games/kouleslib/creator1.raw
/usr/lib/games/kouleslib/colize.raw
/usr/lib/games/kouleslib
/usr/games/koules
以上只是几个常见例子。随着您进一步的使用RPM，您会发现它的各种功能选项组合可以实现更为强大的RPM包管理功能。

06.使用技巧集六

我们一起在风雨中前行

RPM不仅在安装,升级,卸载方面工作出色,而且在查询方面比其它软件包管理工具更胜一筹。这从以下几种情况可以看出:

* 当你在浏览系统文件时,发现一个文件,想知道它来自哪个软件包时,可以用RPM来查询得知;
* 当你的朋友给你发送来一个软件包,但你不知道这是个什么样的软件包,不知道它做些什么,安装些什么,来源是哪里。这时,你可以用RPM查询搞定;
* 几个月前你安装了XFree86窗口软件,但现在你忘了它的版本号,也不知它的说明文档在哪里。这时,你可以用RPM查询一下这个软件包,得到这方面的信息。

RPM的查询还有一个高级功能,即定制输出功能。你可用--queryformat(或-qf)来定制一下输出格式,这样,RPM查询得到的信息将以你定制的格式输出,这样很是方便,尤适合于程序的自动处理。

命令格式
查询RPM格式的软件包，可使用如下命令格式：
rpm -q [查询选项1 查询选项2...]
注：也可使用--query代替-q，效果相同。
选项列表
选项详解
指定用选项中ftp与http相关的四个选项(--ftpproxy,--ftpport,--httpproxy,--httpport)和通用选项的解释,请参见<<精通RPM之二--安装篇>>,本文不再赘述。下面对指定用选项做些解释:

指定用选项可分为如下几类:
　　一、软件包选择类
此类选项在一次只能选择一个,选择多个时RPM将提示错误:
rpm: one type of query/verify may be performed at a time
从查询方面看,一个是查询那些已安装的软件包,一个是查询未安装的软件包。
　　1. 查询已安装的软件包,使用下列选项:
　　(1) -a (或--all) : 查询所有已安装的软件包

```
# rpm -q -a
setup-2.0.2-1
filesystem-1.3.4-5
basesystem-6.0-5
agrep-2.04-5
aktion-0.3.6-2
amor-0.5-1
dhcpcd-1.3.17pl2-1
ldconfig-1.9.5-15
glibc-2.1.2-12
chkconfig-1.0.6-2
......
#
```

注: 本例查找当前系统中安装的所有软件包,输出很多,仅列出几个,剩下的省略掉了(以......表示)。

　　(2) -g (或--group) : 查询有哪些软件包属于指定类别
RPM根据软件功用的不同,将软件分为以下若干类:(括号内为注释)
Amusements/Games (娱乐/游戏)
Amusements/Graphics(娱乐/图形)
Applications/Archiving (应用/档案)
Applications/Communications (应用/通讯)
Applications/Databases (应用/数据库)
Applications/Editors (应用/编辑器)
Applications/Emulators (应用/仿真器)
Applications/Engineering (应用/工程)
Applications/File (应用/文件)
Applications/Internet (应用/因特网)
Applications/Multimedia (应用/多媒体)

Applications/Productivity (应用/产品)
Applications/Publishing (应用/印刷)
Applications/System (应用/系统)
Applications/Text (应用/文本)
Development/Debuggers (开发/调试器)
Development/Languages (开发/语言)
Development/Libraries (开发/函数库)
Development/System (开发/系统)
Development/Tools (开发/工具)
Documentation (说明文档)
System Environment/Base (系统环境/基础)
System Environment/Daemons (系统环境/守护)
System Environment/Kernel (系统环境/内核)
System Environment/Libraries (系统环境/函数库)
System Environment/Shells (系统环境/接口)
User Interface/Desktops (用户界面/桌面)
User Interface/X (用户界面/X窗口)
User Interface/X Hardware Support (用户界面/X硬件支持)
Other (其它)

注意: 类别是大小写敏感的,这一点输入时要小心。如果用户想要查询当前系统安装了哪些游戏类的软件包,可这样做:

```
# rpm -q -g Amusements/Games
clanbomber-1.01-1
kdegames-1.1.2-1
xbill-2.0-6
xboard-4.0.0-3
xboing-2.4-7
xgammon-0.98-14
xjewel-1.6-11
xpat2-1.04-10
xpilot-3.6.2-6
xpuzzles-5.4.1-7
xtrojka-1.2.3-6
#
```

如果用户想查询若干类别的软件包时,可以把类别都列出来,一起查询,如:

```
# rpm -q -g Applications/Editors Applications/File
emacs-20.3-15
emacs-nox-20.3-15
emacs-X11-20.3-15
vim-common-5.6-12
vim-minimal-5.6-12
vim-X11-5.6-12
fileutils-4.0-3
bzip2-0.9.5d-1
findutils-4.1-32
```

git-4.3.17-6
gzip-1.2.4-15
slocate-1.4-7
stat-1.5-12
file-3.26-6
#
注: 本例查询结果中前6个为"应用/编辑器"类别,后8个为"应用/文件"类别。

(3) -f (或--file) : 查询有哪些软件包拥有指定文件这个选项非常有用。当用户不清楚某个文件属于哪个软件包时,可试试这个选项。假如我记不清/usr/sbin/ftpd这个文件在哪个包中,现在用RPM查询:

rpm -qf /usr/sbin/ftpd
inet6-apps-0.36-3
#
现在知道了,它在inet6-apps-0.36-3包中。命令中-qf是-q -f的缩写,当选项仅带一个减号时,多个选项可以合并在一起,如-a
-b -c三个选项可写作-abc,执行效果相同。
(4) --whatrequires : 查询有哪些包需要指定功能
本选项指示RPM从所有已安装的软件包中,查询有哪些软件包提供了用户指定的功能,命令行上可输入一个或多个功能。
rpm -q --whatrequires /bin/ps libc.so.6 | head
autofs-3.1.3-2
agrep-2.04-5
aktion-0.3.6-2
amor-0.5-1
chkconfig-1.0.6-2
libtermcap-2.0.8-16
bash-2.03-10
ncurses-4.2-18
info-3.12f-4
fileutils-4.0-3
#
注: 本例从系统中查询哪些软件包需要/bin/ps和libc.so.6功能,通过管道输出前10行内容。
(5) --whatprovides : 查询有哪些包提供指定功能
本选项指示RPM从所有已安装的软件包中,查询有哪些软件包提供了用户指定的功能。命令行上可输入一个或多个功能。
rpm -q --whatprovides libc.so.6 /bin/ps
glibc-2.1.2-12
no package provides /bin/ps
#

注: 本例从系统中查询有哪些软件包提供libc.so.6与/bin/ps功能,结果是glibc-2.1.2-12包提供了libc.so.6,而没有包提供/bin/ps功能。

(6) --triggeredby : 查询有哪些包被指定的包触发本选项指示RPM从所有已安装的软件包中,查询

有哪些包可被用户指定的包触发。指定的包可输入一个或多个。

```
# rpm -q --triggeredby file
foo-8.0-1
#
```

注：本例查询后得知,安装或卸载foo-8.0-1包,将会触发与file包相关的脚本程序执行。

2. 查询未安装的软件包：(仅有一个选项可用)

(1) -p : 查询指定包裹文件的信息

使用本选项,通过指定一个或多个包裹文件名,RPM就可以获取相应包裹的有关信息。包裹文件可以是一般形式,也可是ftp/http形式。

```
# rpm -qp -l --ftpport 2121 ftp://zzz:pass@linux/zhsoft/file-3.26-6.i386.rpm

/usr/bin/file
/usr/man/man1/file.1
/usr/man/man4/magic.4
/usr/share/magic
#
```

注：本例查询ftp形式的包裹,显示其文件列表(-l选项使然,下面要讲到)。ftp用户名zzz,密码为pass,远程机为linux,文件名为/zhsoft/file-3.26-6.i386.rpm,ftp使用2121端口。

二、信息显示类

本类选项可同时选择多个,以显示多种信息。

1. -i : 显示软件包信息

当用户需要了解软件包的头部信息时,要使用该选项。

```
# rpm -qi file
Name: file Relocations: /usr
Version : 3.26 Vendor: Red Hat Software
Release : 6 Build Date: 1999年03月23日 星期二 05时32分29秒
Install date: 2001年10月14日 星期日 21时18分25秒 Build Host: porky.devel.redhat.com

Group : Applications/File Source RPM: file-3.26-6.src.rpm
Size: 211946 License: distributable
Packager: Red Hat Software
Summary : A utility for determining file types.
Description :
The file command is used to identify a particular file according to the

type of data contained by the file. File can identify many different
file types, including ELF binaries, system libraries, RPM packages, and

different graphics formats.
You should install the file package, since the file command
is such a
useful utility.
#
```

上述输出中,包含多个域,各个域的含义为:
Name : 软件名;Relocations : 重定位目录前缀(一个或多个);
Version : 版本号;Vendor : 开发商;
Release : 释出号;Build Date : 包建立时间;
Install date : 安装时间; Build host : 包建立主机;
Group: 类别; Source RPM : 源代码包;
Size : 大小; License : 许可证;
Packager : 打包者;
Summary : 软件简介;
Description : 详细描述。

2. -l (或--list) : 显示软件包文件列表

当用户想知道软件包包含哪些文件时,要使用该选项。输出时,一行一个文件名,形成文件列表。下例列出file包中文件:

```
# rpm -qlv file
-rwxr-xr-x root root23948 3月 23 1999 /usr/bin/file
-rwxr-xr-x root root12023 3月 23 1999 /usr/man/man1/file.1
-rwxr-xr-x root root 6625 3月 23 1999 /usr/man/man4/magic.4
-rwxr-xr-x root root 169350 3月 23 1999 /usr/share/magic
#
```

注: 本例列文件时选用了通用选项-v,这样列出的格式同ls命令列出的格式,从中可以看到各个文件的权限,属主,属组,大小,最后修改时间及文件名等信息。

3. -d (或--docfiles) : 显示软件包说明文档RPM将软件包中的文件分为三类,一是配置文件,一是说明文档,再一是其它文件(包括可执行程序及数据文件等)。通过-d选项,可列出包中所有说明文档。下例列出file包中说明文档:

```
# rpm -qd file
/usr/man/man1/file.1
/usr/man/man4/magic.4
#
```

4. -c (或--configfiles) : 显示软件包配置文件使用本选项可列出包中所有配置文件。下例列出lze包中的配置文件:

```
# rpm -qc lze
/etc/funkey.def
/etc/inputme.def
#
```

5. -s (或--state) : 显示软件包文件状态RPM已安装的软件包中各个文件可拥有如下4种状态:

* normal(正常): 这表明文件未被其它软件包修改过。
* replaced(已替换): 这表明文件已被其它软件包修改替换过了,不再是原先的文件了。* not installed(未安装): 这表明文件未安装。有一种原因可导致这种情况发生,这就是当安装软件包时使用

了--excludedocs选项(或是通过设置%_excludedocs宏为1),说明文档未被安装到系统中。当用RPM查询此软件包状态时,所有说明文档均显示此状态。

* net shared(网络共享): 这表明文件处于网络共享状态。这一状态是RPM用来支持NFS(网络文件系统)的,用于避免一个NFS客户端系统删除一个正在共享的文件时,另一NFS客户端系统无法正常使用含此文件的软件包。有两种情况可使文件在此状态,一是文件安装在真实网络共享的目录里,二是文件安装在RPM资源配置文件中%_netsharedpath宏所确定的目录里。

```
# rpm -i --excludedocs file-3.26-6.i386.rpm
# rpm -qs file
normal/usr/bin/file
not installed /usr/man/man1/file.1
not installed /usr/man/man4/magic.4
normal/usr/share/magic
#
```

注: 本例先安装file包裹文件,选用--excludedocs表明不安装说明文档。而后再查询file包文件的状态,可以看到其说明文档均为not
installed(未安装),其它文件为normal(正常)状态。

6. -R (或--requires) : 显示软件包所需的功能

所谓功能,可以是软件包标识,可以是文件(如共享库libc.so.6等),也可以是虚拟的名字。软件包的依赖关系,就是依靠功能来实现的。RPM安装软件包时,如果所需功能不存在,则其依赖关系不满足,RPM将中断安装过程。

```
# rpm -q -R lze
/bin/sh
ld-linux.so.2
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
#
```

注: 本例查询lze包所需的功能。

7. --provides : 显示软件包提供的功能

一个软件包,可以提供若干功能,这些功能可以是动态链接库等实际的文件,也可以是虚拟的名字(只要其它软件包可以用就行了)。如果想查询某个包提供了哪些功能,要用--provides

选项。

下面的例子查询一个包裹文件所提供的功能:

```
# rpm -qp --provides zlib-1.1.3-6.i386.rpm
libz.so.1
libz.so.1(GCC.INTERNAL)
#
```

注: 本包裹文件提供的功能是libz.so.1动态链接库。

8. --conflicts : 显示软件包冲突的功能

何谓冲突?冲突就是不同软件包之间不能共存的现象。RPM制作软件包时,可以记录下与本软件包不能共存的功能,这样安装该包时,若冲突的功能已然存在,则RPM将中止安装。

下面的例子查询at-3.1.7-8包是否有与其冲突的功能:

rpm -q --conflicts at-3.1.7-8
crontabs <= 1.5
#
注: 本例显示at包与版本号小于1.5的crontabs包有冲突存在。

9. --scripts : 显示软件包内置的脚本程序

scripts即脚本,指的是用SHELL语言编写的程序。选用本选项时,RPM将输出软件包所含的各类脚本程序的内容。总共有5类这样的脚本程序,即安装前脚本程序(preinstall),安装后脚本程序(postinstall),卸载前脚本程序(preuninstall),卸载后脚本程序(postuninstall)和校验脚本程序(verify)。

下面的例子列出zsh包的脚本程序:

```
# rpm -q --scripts zsh-3.0.7-4 | nl
1 postinstall script (through /bin/sh):
2 if [ ! -f /etc/shells ] ; then
3 echo "/bin/zsh" > /etc/shells
4 else
5 echo "/bin/zsh" >> /etc/shells
6 fi
7 /sbin/install-info /usr/info/zsh.info.gz /usr/info/dir \
8 --entry="* zsh: (zsh). An enhanced bourne shell."
9 preuninstall script (through /bin/sh):
10 if [ "$1" = 0 ] ; then
11 /sbin/install-info --delete /usr/info/zsh.info.gz /usr/info/dir \
12 --entry="* zsh: (zsh). An enhanced bourne shell."
13 fi
14 postuninstall script (through /bin/sh):
15 if [ "$1" = 0 ] ; then
16 if [ -f /etc/shells ] ; then
17 TmpFile=`/bin/mktemp /tmp/.zshrpmXXXXXX`
18 grep -v '^/bin/zsh$' /etc/shells > $TmpFile
19 cp -f $TmpFile /etc/shells
20 rm -f $TmpFile
21 chmod 644 /etc/shells
22 fi
23 fi
#
```

注: 本例利用管道技术给查询的每行输出加上了行号,便于观察解释。
第2-8行: 为安装后脚本程序(postinstall)的源代码;
第10-13行: 为卸载前脚本程序(preuninstall)的源代码;
第15-23行: 为卸载后脚本程序(postuninstall)的源代码。
例中所有脚本程序均通过/bin/sh解释执行,本例没有安装前脚本程序(preinstall)和
校验脚本程序(verify)。

10. --triggers : 显示软件包内置的触发脚本程序

触发脚本程序是scripts类脚本程序的扩展,它用于软件包之间的交互控制。触发脚本
程序有安装时触发脚本程序(triggerin),卸载前触发脚本程序(triggerun)和卸载后触发脚
本程序(triggerpostun)三种。
下面的例子列出zsh包中的触发脚本程序:
rpm -q --triggers zsh-3.0.7-4

```
triggerpostun script (through /bin/sh) -- zsh <= 3.0.7-2
if [ ! -f /etc/shells ] ; then
echo "/bin/zsh" > /etc/shells
else
echo "/bin/zsh" >> /etc/shells
fi
#
```

注: 输出的第1行说明脚本程序的类别(为卸载后触发脚本程序triggerpostun),用什么程序解释(一般为/bin/sh),和触发的条件(在--之后描述)。本例的触发条件为zsh的版本小于或等于3.0.7-2。如果条件成立,则此脚本程序将会执行。
输出的第2-6行为卸载后触发脚本程序的源代码。

11. --changelog : 显示软件维护记录

changelog即软件维护记录,用它来记录什么人,什么时间,改动了软件的什么地方。通过查看维护记录,开发者或用户可以了解软件的开发进展情况。
下面的例子查询lze包的维护情况:

```
# rpm -q --changelog lze-6.0-2
* 五 5月 01 1998 雨亦奇
- 增加多窗口操作
* 一 3月 24 1997 雨亦奇
- 增加块操作命令
#
```

注: 从本例中就可以看出来,维护记录的书写有一定的规范:以星号(*)开头的行记录维护时间,维护人及其电子邮箱,而以减号(-)开头的行则记录维护的具体内容。

12. --dump : 显示软件包中所有文件的属性信息

如果用户想查看某个软件包内文件的属性信息,请用--dump选项。

```
# rpm -q --dump file
/usr/bin/file 23948 922138347 abaf6cfd51588ac7c484526fbdb84e5b 0100755
root root 0 0 0 X
/usr/man/man1/file.1 12023 922138346 76d9ff6567ab64a53eab50911272f5c1
0100755 root root 0 1 0 X
/usr/man/man4/magic.4 6625 922138346 b8d126803709f0da7f39f5125a132cd3
0100755root root 0 1 0 X
/usr/share/magic 169350 922138346 3bd2eaf3c5e0e84153ba7df38b7561fc 0100755
root root 0 0 0 X
#
```

现在根据例子的输出,解释一下RPM包中各个文件所拥有的属性信息:(以第1行为例)
/usr/bin/file : 为文件名;
23948 : 指文件大小(字节数);
922138347 : 指文件最后修改时间(秒数);
abaf6cfd51588ac7c484526fbdb84e5b : 指根据文件内容计算出的MD5检查和;
0100755 : 指文件存取权限;
root : 指文件属主;

root : 指文件属组;
0 : 配置文件标志,为0表示该文件非配置文件,为1表明该文件是配置文件;
0 : 说明文档标志,为0表示该文件非说明文档,为1表明该文件是说明文档;
0 : 指文件类型,为0表示普通文件,非0表示设备文件,包含主从设备号;
X : 符号连接内容,如果文件不是符号连接,则此值为X。

三、定制输出类

定制输出是RPM一项强有力的功能,当用户想要RPM按自己的格式输出软件包信息时,可以通过使用--qf (或--queryformat)选项来实现定制。

定制输出要指定格式化的字符串(类似printf的格式串),该字符串以单引号'或双引号"引住。格式串中可包括普通文本,含转义符的文本,功能标签和数组循环三种不同的组件。

* 普通文本
格式串中的普通文本将按原样输出。
* 含转义符的文本
RPM定义的转义符为\,称作反斜杠。当RPM遇到此字符时,会根据预先定义的转义序列,把后面的字符解释后输出。
转义序列如下:
\a : 输出响铃字符(值为7),效果是喇叭鸣叫一声。
\b : 输出退格符(值为8),效果是删除光标前面一个字符,且光标后退一个字符位置。
\f : 输出换页符(值为12),根据终端解释的不同,效果也不相同,有的是实现清屏操作,有的则是换一行。
\n : 输出换行符(值为10),效果是光标移到下一行。
\r : 输出回车符(值为13),效果是光标回到行首。
\t : 输出跳格(值为9),根据终端解释的不同,效果也不同,有的是输出一个空格,有的 则输出最多8个空格。
\v : 输出垂直方向的跳格(值为11),用处不大。
\\ : 输出反斜杠(\)这个字符。

* 功能标签
RPM内置了很多功能标签,如NAME表示软件名,VERSION表示版本号,RELEASE表示释出号等。输出功能标签所表示的内容时,需用这样一种格式:

%[输出宽度]{功能标签}
注:输出宽度可选,正值表示右对齐输出,负值表示左对齐输出。
如格式串中使用%{NAME}时将输出软件名,使用%20{NAME}时则右对齐输出软件名,宽度为20,而使用%-20{NAME}时则按宽度为20左对齐输出软件名。看看下面的实际例子就会明白:

```
# rpm -q --qf "%{NAME}:%20{NAME}:%-20{NAME}:\n" file
file:file:file:
#
```

功能标签书写不区分大小写,即NAME也可输作name,Name等等。功能标签还可以带有选项,选项用":选项名"表示,选项名有大小写之分。如%{FILEMODES}默认以数字形式输出,如采用perms选项,即%{FILEMODES:perms},输出结果将采用rwx的形式。请看下面的例子:

```
# rpm -q --qf "%{filenames} %{FILEMODES:perms}\n" file
/usr/bin/file -rwxr-xr-x-
#
```

注：%{filenames}表示包中的文件名。

RPM常用功能标签表

注：可用rpm --querytags命令查到RPM内置的所有功能标签。

* 数组循环

对于功能标签是数组的,可以用数组循环(用[与]括住的部分)来输出数组的全部内容。数组循环内部可包含功能标签,普通文本和含转义符的文本。RPM将根据循环内部一个或多个数组类型的功能标签所拥有的单元个数,取其最小值,循环同样次数,输出解释或转义后的数据。对于功能标签仅含一个单元的,若想输出多次,则需要在功能标签前面加个等号(=)。

下面的例子以数组循环的方式,输出file包所含的各个文件的文件名及权限,同时输出每一个文件时都要先输出软件包标识。

```
# rpm -q --qf "[%{=NAME}-%{=VERSION}-%{=RELEASE} : %-20{FILENAMES}
: %{FILEMODES:perms}\n]" file
file-3.26-6 : /usr/bin/file : -rwxr-xr-x-
file-3.26-6 : /usr/man/man1/file.1 : -rwxr-xr-x-
file-3.26-6 : /usr/man/man4/magic.4 : -rwxr-xr-x-
file-3.26-6 : /usr/share/magic : -rwxr-xr-x-
#
```

通过使用定制输出,我们可以很方便地查找出系统中占用空间最大的软件包。

```
# rpm -q -a --qf "%{NAME} %{SIZE}\n" | sort -nr +1 | head -1

kernel-source 65824967
#
```

注：-a项选择所有已安装的软件包,--qf项定制查询格式, 格式串中用功能标签NAME输出软件名,用SIZE输出软件大小,查询得到的数据通过管道送sort命令,对第2列降序排序后由head命令取第1行内容,这即是占用空间最大的软件包。

07.RPM命令手册

我们一起在风雨中前行

一、安装

命令格式：
rpm -i (or --install) options file1.rpm ... fileN.rpm
参数：
file1.rpm ... fileN.rpm 将要安装的RPM包的文件名
详细选项：
-h (or --hash) 安装时输出hash记号 (``#'')
--test 只对安装进行测试，并不实际安装。
--percent 以百分比的形式输出安装的进度。
--excludedocs 不安装软件包中的文档文件
--includedocs 安装文档
--replacepkgs 强制重新安装已经安装的软件包
--replacefiles 替换属于其它软件包的文件
--force 忽略软件包及文件的冲突
--noscripts 不运行预安装和后安装脚本
--prefix 将软件包安装到由 指定的路径下
--ignorearch 不校验软件包的结构
--ignoreos 不检查软件包运行的操作系统
--nodeps 不检查依赖性关系
--ftpproxy 用 作为 FTP代理
--ftpport 指定FTP的端口号为
通用选项
-v 显示附加信息
-vv 显示调试信息
--root 让RPM将指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile 设置rpmrc文件为
--dbpath 设置RPM 资料库存所在的路径为
二、删除
命令格式：
rpm -e (or --erase) options pkg1 ... pkgN
参数
pkg1 ... pkgN ：要删除的软件包
详细选项
--test 只执行删除的测试
--noscripts 不运行预安装和后安装脚本程序
--nodeps 不检查依赖性
通用选项
-vv 显示调试信息
--root 让RPM将指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile 设置rpmrc文件为
--dbpath 设置RPM 资料库存所在的路径为
三、升级
命令格式
rpm -U (or --upgrade) options file1.rpm ... fileN.rpm

参数

file1.rpm ... fileN.rpm 软件包的名字

详细选项

-h (or --hash) 安装时输出hash记号 (``#'')

--oldpackage 允许"升级"到一个老版本

--test 只进行升级测试

--excludedocs 不安装软件包中的文档文件

--includedocs 安装文档

--replacepkgs 强制重新安装已经安装的软件包

--replacefiles 替换属于其它软件包的文件

--force 忽略软件包及文件的冲突

--percent 以百分比的形式输出安装的进度。

--noscripts 不运行预安装和后安装脚本

--prefix 将软件包安装到由 指定的路径下

--ignorearch 不校验软件包的结构

--ignoreos 不检查软件包运行的操作系统

--nodeps 不检查依赖性关系

--ftpproxy 用 作为 FTP代理

--ftpport 指定FTP的端口号为

通用选项

-v 显示附加信息

-vv 显示调试信息

--root 让RPM将指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下

--rcfile 设置rpmrc文件为

--dbpath 设置RPM 资料库存所在的路径为

四、查询

命令格式：

rpm -q (or --query) options

参数：

pkg1 ... pkgN ：查询已安装的软件包

详细选项

-p (or ``-'') 查询软件包的文件

-f 查询属于哪个软件包

-a 查询所有安装的软件包

--whatprovides 查询提供了 功能的软件包

-g 查询属于 组的软件包

--whatrequires 查询所有需要 功能的软件包

信息选项

显示软件包的全部标识

-i 显示软件包的概要信息

-l 显示软件包中的文件列表

-c 显示配置文件列表

-d 显示文档文件列表

-s 显示软件包中文件列表并显示每个文件的状态

--scripts 显示安装、卸载、校验脚本

--queryformat (or --qf) 以用户指定的方式显示查询信息

--dump 显示每个文件的所有已校验信息

--provides 显示软件包提供的功能
--requires (or -R) 显示软件包所需的功能
通用选项
-v 显示附加信息
-vv 显示调试信息
--root 让RPM将指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile 设置rpmrc文件为
--dbpath 设置RPM 资料库存所在的路径为
五、校验已安装的软件包
命令格式：
rpm -V (or --verify, or -y) options
参数
pkg1 ... pkgN 将要校验的软件包名
软件包选项
-p Verify against package file
-f 校验所属的软件包
-a Verify 校验所有的软件包
-g 校验所有属于组 的软件包
详细选项
--noscripts 不运行校验脚本
--nodeps 不校验依赖性
--nofiles 不校验文件属性
通用选项
-v 显示附加信息
-vv 显示调试信息
--root 让RPM将指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
--rcfile 设置rpmrc文件为
--dbpath 设置RPM 资料库存所在的路径为

六、校验软件包中的文件
语法：
rpm -K (or --checksig) options file1.rpm ... fileN.rpm
参数：
file1.rpm ... fileN.rpm 软件包的文件名
Checksig--详细选项
--nopgp 不校验PGP签名
通用选项
-v 显示附加信息
-vv 显示调试信息
--rcfile 设置rpmrc文件为

七、其它RPM选项
--rebuilddb 重建RPM资料库
--initdb 创建一个新的RPM资料库
--quiet 尽可能的减少输出
--help 显示帮助文件
--version 显示RPM的当前版本

08.RPM升级篇

我们一起在风雨中前行

升级做什么

RPM的升级功能是它受到用户好评的原因之一。因为用户自己将一个软件包从旧版本升级到新版本,特别是大型软件,需要有经验的支持和技术的积累,比较复杂,而用RPM升级软件,只需一个rpm
-U命令就可以了,极大方便了用户。

软件升级基本做两项工作,一是安装新版本,二是卸载旧版本。RPM还有一项重要的工作要做,这就是妥善处理配置文件(CONFIG FILE)。若直接采用安装方式,则用户已配置好的配置文件就会被覆盖,不符合用户要求。

配置文件处理

RPM对某个配置文件,通过比较三种不同的MD5检查和(checksum)来决定如何处理它。这三种不同的MD5检查和是:

1. 原检查和。它是旧版本软件包安装时配置文件的MD5检查和。
2. 当前检查和。它是升级时旧版本配置文件的MD5检查和。
3. 新检查和。它是新版本软件包中配置文件的MD5检查和。

RPM针对以下几种情况分别处理:

1. 当原检查和=X,当前检查和=X,新检查和=X时:

这表明配置文件未曾修改过。此时,RPM会将新的配置文件覆盖掉原文件,而不是对原文件不作处理,原因在于: 虽然文件名和文件内容都没有变化,但文件别的方面的属性(如文件的属主,属组,权限等)却可能改变,所以有必要覆盖一下。

2. 当原检查和=X,当前检查和=X,新检查和=Y时:

这表明原配置文件没有改动过,但是它与新软件包中的配置文件却有所不同。这种情况下,RPM将用新文件覆盖掉旧文件,并且旧文件不作保存(因为它不曾改动过,没有必要保存)。

3. 当原检查和=X,当前检查和=Y,新检查和=X时:

这表明新文件与旧文件内容相同,但当前文件已经作过修改,这些修改对于新版本来说应该是合法的,可以使用的。因此,RPM对当前文件予以保留。

4. 当原检查和=X,当前检查和=Y,新检查和=Y时:

这表明原文件经过修改,现在已与新文件相同,这或许是用户用来修补安全上的漏洞,新版本也作了同样的修改。这种情况下,RPM将新文件覆盖当前文件,避免文件属性方面的不同。

5. 当原检查和=X,当前检查和=Y,新检查和=Z时:

这表明用户已修改了原文件,并且当前内容与新文件内容不同。这种情况下,RPM无法保证新版本软件能正常使用当前的配置文件,所以采用了一个比较明智的办法,既能保护用户的配置数据,又能保证新版本软件正常。这种作法就是将当前文件换名保存(给原文件名加个.rpmsave的后缀,如原文件名为ABC,则换名后为ABC.rpmsave),同时安装新文件,并给出警告信息,如:

warning: /etc/.funkey saved as /etc/.funkey.rpmsave
6. 当没有原检查和时:
此种情况下,当前检查和与新检查和已无关紧要,这表明没有安装过此配置文件。因为没有安装过此配置文件,所以RPM无法判断当前文件是否被用户修改过。这种情况下,RPM会将当前文件换名保存(原文件名后缀不是加个.rpmsave,而是.rpmorig),同时安装新文件,并给出警告信息,如:

warning: /etc/.inputdef saved as /etc/.inputdef.rpmsave
升级命令格式
升级RPM包时,请用以下命令格式:
rpm -U [升级选项1 升级选项2...] [软件包标识1 软件包标识2...]
其中: 也可使用--upgrade代替-U,效果相同。
软件包标识
有关软件包标识的定义,请参见<<精通RPM之三--卸载篇>>。
选项列表
选项说明
因为升级也是一种安装,所以升级的选项列表与安装选项列表基本相同,只是升级的选项列表增加了一项--oldpackage。现着重说明一下这个选项,其它选项说明见<<精通RPM之二--安装篇>>,在此恕不赘述。

--oldpackage选项: 从名字上就可以看出来是老版本软件包的意思。为什么要将软件"升级"到老版本?(这里的升级其实是降级)这里面有个原因。用户一直好好地用着老版本的软件,当有一天发现有新版本发布时,马上用rpm
-U命令升级到系统中,但因为新版本有"臭虫",所以这个软件暂时不能正常工作。而这时,直接用rpm -U命令是升级不到老版本的,因为一般情况的升级是将老版本升级到新版本,RPM默认这一点。若想升级到老版本,则必须用这个特殊的选项。下面举个例子:

rpm -U -v lze-6.0-1.i386.rpm
package lze-7.0-1 (which is newer then lze-6.0-1) is already installed

#

注: 本例在升级过程中出现错误,RPM提示lze软件包已安装,并且现存版本号7.0,高于准备升级的版本号6.0,升级无法继续。
若在命令行使用--oldpackage,结果会怎么样呢?
rpm -U -v --oldpackage lze-6.0-1.i386.rpm
lze-6.0-1
#
注: 命令执行后输出了软件包标识lze-6.0-1,表明升级到老版本成功了。
下面通过输出调试信息来观察一下升级软件包时RPM做的主要工作:
rpm -U -vv --oldpackage foo-3.0-2.i386.rpm 2>&1 | nl
1 D: counting packages to install
2 D: found 1 packages
3 D: looking for packages to download

```
4 D: retrieved 0 packages
5 D: New Header signature
6 D: Signature size: 68
7 D: Signature pad : 4
8 D: sigsize : 72
9 D: Header + Archive: 1577
10 D: expected size : 1577
11 D: opening database mode 0x42 in //var/lib/rpm/
12 D: found 0 source and 1 binary packages
13 D: requires: /bin/sh satisfied by db file lists.
14 D: installing binary packages
15 D: getting list of mounted filesystems
16 D: New Header signature
17 D: Signature size: 68
18 D: Signature pad : 4
19 D: sigsize : 72
20 D: Header + Archive: 1577
21 D: expected size : 1577
22 D: package: foo-3.0-2 files test = 0
23 D:file: /etc/foo.conf action: create
24 D:file: /usr/bin/foo action: create
25 D: running preinstall script (if any)
26 + echo preinstall
27 preinstall
28 foo-3.0-2
29 D: running postinstall scripts (if any)
30 + echo postinstall
31 postinstall
32 + echo triggerinstall
33 triggerinstall
34 + echo triggeruninstall
35 triggeruninstall
36 + echo preuninstall
37 preuninstall
38 D: will remove files test = 0
39 D:file: /usr/bin/foo action: skip
40 D:file: /etc/foo.conf action: skip
41 D: running postuninstall script (if any)
42 + echo postuninstall
43 postuninstall
44 D: removing database entry
45 D: removing name index
46 D: removing group index
47 D: removing requiredby index for /bin/sh
48 D: removing trigger index for file
49 D: removing trigger index for file
50 D: removing trigger index for file
```

51 D: removing file index for foo.conf
52 D: removing file index for foo

注: 第1-4行: 计算命令行上要升级的包数,并且下载那些需要下载的包裹文件;
第5-10行: 根据包裹文件头部信息,确定软件占用空间;
第11,12行: 打开RPM数据库及包裹文件;
第13行: 检查依赖是否满足,本例满足;
第14行: 安装执行程序包;
第15行: 取当前已安装文件系统列表;
第16-21行: 再度检查包裹头信息,确定占用系统空间;
第22-24行: 确定包中各个文件的执行操作(action),均为建立(create);
第25行: 执行安装前脚本程序(如果有的话);
第26-27行: 以+开头的为脚本程序执行的命令,其后为其输出结果;
第28行: 安装foo-3.0-2包;
第29行: 执行安装后脚本程序(如果有的话);
第30-31行: 以+开头的为脚本程序执行的命令,其它为执行结果;
第32-33行: 执行安装时触发脚本程序;
第34-35行: 执行卸载前触发脚本程序,自此开始卸载原软件包;
第36-37行: 执行卸载前脚本程序;
第38-40行: 确定原包中各文件的执行操作,本例均为跳过(skip),即不作处理;
第41-43行: 执行卸载后脚本程序;
第44-52行: 删除原包在RPM数据库中的所有信息(数据及索引)。

09.RPM之安装篇

我们一起在风雨中前行

从一般意义上说，软件包的安装其实是文件的拷贝，RPM安装软件包，也无外乎此。但RPM要更进一步、更聪明一些就需要多做些工作了。

聪明的安装

从一般意义上说，软件包的安装其实是文件的拷贝，即把软件所用的各个文件拷贝到特定目录。RPM安装软件包，无外乎此。但RPM要更进一步，更聪明一些。在安装前，它通常要执行以下操作：

1.检查软件包的依赖（Dependency）

RPM格式的软件包中可包含有依赖关系的描述，如软件执行时需要什么动态链接库，需要什么程序存在及版本号要求等。当RPM检查时发现所依赖的链接库或程序等不存在或不符合要求时，默认的做法是中止软件包安装。

2.检查软件包的冲突（Conflicts）

有的软件与某些软件不能共存，软件包制作者会将这种冲突记录到RPM软件包中。安装时，若RPM发现有冲突存在，将会中止安装。

3.执行安装前脚本程序（Preinstall）

此类程序由软件包制作者设定，需要在安装前执行。通常是检测操作环境，建立有关目录，清理多余文件等等，为顺利安装作准备。

4.处理配置文件（Configfiles）

RPM对配置文件（Configfiles）有着特别的处理。因为用户常常需要根据实际情况，对软件的配置文件做相应的修改。如果安装时简单地覆盖了此类文件，则用户又得重新手工设置，很麻烦。这种情况下，RPM做得比较明智：它将原配置文件换个名字保存了起来（原文件名后缀加上.rpmorig），用户可根据需要再恢复，避免重新设置的尴尬。

5.解压软件包并存放到相应位置

这是最重要的部分，也是软件包安装的关键所在。在这一步，RPM将软件包解压缩，将其中的文件一个个存放到正确的位置，同时，文件的操作权限等属性相应设置正确。

6.执行安装后脚本程序（Postinstall）

此类程序为软件的正确执行设定相关资源，如修改inetd.conf、运行ldconfig程序以利新的动态链接库生效等等。

7.更新RPM数据库

安装后，RPM将所安装的软件及相关信息记录到其数据库中，便于以后升级、查询、校验和卸载。

8.执行安装时触发脚本程序（Triggerin）

触发脚本程序是指软件包满足某种条件（如已安装软件包sendmail，或file版本大于3.0）时才触发执行的脚本程序，它用于软件包之间的交互控制。触发脚本程序有三类：一是软件包安装时才触发的，称为安装时触发脚本程序（triggerin）；二是软件包卸载前触发的，叫作卸载前触发脚本程序（triggerun）；三是软件包卸载后才触发执行的，称作卸载后触发脚本程序（triggerpostun）。这些触发脚本程序，大大扩展了RPM软件包管理的功能。

命令格式

安装RPM格式的软件包，可使用如下命令格式： rpm -i [安装选项1 安装选项2...] 包裹文件1 包裹文件2...

注：可用--install代替-i，效果相同。

选项列表

包裹文件

对于要安装的RPM格式的包裹文件，RPM对其名字不作强制要求。用户可以使用以下三种方式的命名样式：

1.典型的命名样式（常用）： 格式为：软件名-版本号-释出号.体系号.rpm

注：体系号指的是执行程序适用的处理器体系，如i386体系，sparc体系等。体系号为src时表明为源代码包，否则为执行程序包。

如abc-3.2-1.i386.rpm为执行程序包，软件名为abc，版本号为3.2，释出号为1，适用体系为i386，而abc-3.2-1.src.rpm则为源代码包。

2.URL形式的命名样式（较常用）

*

FTP方式的命名格式：ftp://[用户名[:密码]@]主机[:端口]/包裹文件

注：[]括住的内容表示可选。主机可以是主机名，也可是IP地址。包裹文件可含目录信息。如未指定用户名，则RPM采用匿名方式传输数据（用户名为anonymous）。如未指定密码，则RPM会根据实际情况提示用户输入密码。如未指定端口，则RPM使用默认端口（一般为21）。

如ftp://ftp.xxx.com/yyy.rpm

如ftp://24.109.164.55:1024/pub/yyy.rpm

如zhsoft@ftp.xxx.com/yyy.rpm">ftp://zhsoft@ftp.xxx.com/yyy.rpm

如password@ftp.xxx.com/yyy.rpm">ftp://zhsoft:password@ftp.xxx.com/yyy.rpm

3 HTTP方式的命名：格式为：http://主机[:端口]/包裹文件

注：[]括住的内容可选。主机可以是主机名，也可是IP地址。包裹文件可含目录信息。如未指定端口，则RPM默认使用80端口。

如http://www.xxx.com/yyy.rpm

如http://www.xxx.com:8080/pub/yyy.rpm

3.其它形式（很少使用）：

命名格式：任意

如将abc-3.2-1.i386.rpm改名为abc.txt，用RPM安装也会安装成功，其根本原因是RPM判定一个文件是否RPM格式，不是看名字，而是看内容，看其是否符合特定的格式。

参数：

一、安装指定用选项

1.hash（或-h）：以#显示安装进度,如果一个软件包很大、安装费时时，用户若想及时了解安装进度，必须用此选项。该选项以显示#号表示进度，每个#号表示2%的进度，总共要显示50个#号。下面安装MySQL软件包，如下所示：

```
# rpm -i --hash MySQL-3.22.32-1.i386.rpm
##################################################
#
```

从上看出，软件包安装顺利完成。

2.percent：以%（百分比）显示安装进度

percent的含义是百分比，而此选项的作用就是以百分比（%）来显示安装进度的。同样是安装MySQL软件包，下面的输出就不一样：

```
# rpm -i --percent MySQL-3.22.32-1.i386.rpm
%% 0.000000
%% 0.002600
%% 0.020586
......
%% 100.000000
#
```

上例中省略了很多输出（......表示）。因为如果软件包很大，则用--percent时输出的内容会很多，所以用户最好用--hash选项，以#号来表示安装进度，这样简洁明了。

3.test：安装测试

所谓安装测试，意即并非真正的安装，它不拷贝和建立任何文件。使用本选项的目的在于：检测软件包的依赖关系是否满足，是否存在潜在的冲突等等。

```
# rpm -i --test autofs-3.1.3-2.i386.rpm
error: failed dependencies:
mktemp is needed by autofs-3.1.3-2
#
```

本例进行安装测试时发现了依赖方面的错误，autofs所需要的mktemp包不存在，安装中断。

下面先安装mktemp软件包，再进行autofs安装测试，看情况怎么样：

```
# rpm -i mktemp-1.5-2.i386.rpm
# rpm -i --test autofs-3.1.3-2.i386.rpm
#
```

由上看出，这次测试没有出现什么错误。

4.replacepkgs：替换软件包

为什么要替换软件包？原因可能是系统中的软件包已经破坏了，其中一个或多个文件丢失或损毁。如果用户想修复这个软件包，用直接安装的方法，RPM将报错退出：

```
# rpm -V file
missing/usr/man/man1/file.1
# rpm -i file-3.26-6.i386.rpm
package file-3.26-6 is already installed
#
```

注：本例先用RPM校验命令校验一下file软件包，发现/usr/man/man1/file.1文件丢失。之后用安装命令

安装，RPM提示软件包file-3.26-6已经安装了。（有关RPM校验命令，<<精通RPM之六--校验篇>>中将会有详细的介绍）

如果采用--replacepkgs选项，结果又怎样呢？

```
# rpm -i --replacepkgs file-3.26-6.i386.rpm
# rpm -V file
#
```

看来，RPM成功地替换了原软件包，校验一下该包，发现没有错误输出，所以现在的软件包是完整的。

5.replacefiles：替换文件

RPM是聪明的软件包管理器，它维护着每个已安装软件包的文件信息。如果在安装一个新的软件包时，RPM发现其中某个文件和已安装的某个软件包中的文件名字相同但内容不同，那么RPM就会认为这是一个文件冲突，会报错退出：

```
# rpm -i ff-4.0-2.i386.rpm
file /root/my.a from install of ff-4.0-2 conflicts with file from package
zoo-6.0-1
#
```

注：本例中RPM发现要安装的软件包ff-4.0-2与已安装的软件包zoo-6.0-1中，含有相同的一个文件/root/my.a，但其内容并不相同，所以提示了文件冲突的错误。

如果用户想忽略这个错误，可使用--replacefiles选项，指示RPM发现文件冲突时，直接替换掉原文件即可。注意：除非用户对所冲突的文件有很深的了解，不要轻易替换文件，以免破坏已安装软件包的完整性，确保其能正常运行。

```
# rpm -i --replacefiles ff-4.0-2.i386.rpm
#
```

采用该选项后，软件包能顺利安装了。

另外，说到替换文件，若要安装的软件包中的文件已存在，但此文件并不属于任何软件包，RPM的做法是将文件换名保存（文件名后缀加.rpmorig），并且以警告信息提醒用户。如下所示：

```
# rpm -i foo-6.0-1.i386.rpm
warning: /etc/foo.conf saved as /etc/foo.conf.rpmorig
#
```

6.allfiles：安装所有文件

读者看到此选项，也许要问：难道RPM安装软件包不是安装其中所有的文件吗？ 我的回答是：如果是初次安装的话，RPM确是将包中所有文件全部安装。但是，如果是修复软件包（用--replacepkgs选项），那结果就不一定了。个中原因是：RPM包中有些配置文件可标识为missingok属性（missingok指的是即使丢失，照样OK），这样的包安装后，若这种类型的配置文件被删除，则修复时RPM默认的做法是不再安装这种类型的文件，除非采用--allfiles选项。下面看个实际的例子：

```
# rpm -i foo-6.0-1.i386.rpm
```

ls -l /etc/foo.conf
-rw-r--r-- 1 root root9 Oct 11 09:50 /etc/foo.conf
rm -f /etc/foo.conf
rpm -i --replacepkgs foo-6.0-1.i386.rpm
ls -l /etc/foo.conf
ls: /etc/foo.conf: 文件或目录不存在
rpm -i --replacepkgs --allfiles foo-6.0-1.i386.rpm
ls -l /etc/foo.conf
-rw-r--r-- 1 root root9 Oct 11 09:50 /etc/foo.conf
注：本例中已预先知道foo包中的配置文件/etc/foo.conf带有missingok属性。
（1）用rpm -i命令安装foo包；
（2）用ls命令列一下属于该包的配置文件/etc/foo.conf（能列出来，表明文件存在）；
（3）用rm命令删除了这个文件；
（4）用rpm -i --replacepkgs命令修复foo软件包；
（5）因为修复时未用--allfiles选项，所以用ls命令列文件/etc/foo.conf时出错了：文件不存在，没有安装上；

（6）用rpm -i --replacepkgs --allfiles命令安装修复foo软件包；
（7）再度用ls命令列文件，列了出来，表明这次安装上了。

由本例看出，若用户确实想安全恢复（修复）某个软件包，最好使用--allfiles选项。一般情况下则不必这么做，因为带有missingok属性的配置文件本来就是可以丢失的嘛，不必太在意了。

7.force：强制执行

force的含义是强制。--force选项的作用就是强制安装软件包，不考虑软件包是否已安装，也不考虑有没有文件冲突。其效果相当于同时选用--replacepkgs与--replacefiles选项进行安装。

8.excludedocs：不安装说明文档

RPM有多个好功能，其中之一就是：它将文件分为配置文件，说明文档和其它文件三种，这样便于区别对待，灵活处理。

基于RPM安装的LINUX发行版中包括5000多个说明文档，有50M字节，占用的空间不小。如果想节省空间，可使用--excludedocs选项以排除安装说明文档。

如软件包file中包含/usr/man/man1/file.1和/usr/man/man4/magic.4两个说明文档。安装时若使用--excludedocs选项，这两个文档就不会被安装。

rpm -i --excludedocs file-3.26-6.i386.rpm
ls -l /usr/man/man1/file.1
ls: /usr/man/man1/file.1: 文件或目录不存在
ls -l /usr/man/man4/magic.4
ls: /usr/man/man4/magic.4: 文件或目录不存在

#
如果用户想让RPM默认不安装说明文档，则可以这么做：编辑~/.rpmmacros文件（用户主目录HOME下的RPM宏文件），加入下面一行：

%_excludedocs 1
其作用是定义RPM内部的_excludedocs宏为1，确认不安装说明文档。
这样的话，就不用在命令行使用--excludedocs选项了。
rpm -i file-3.26-6.i386.rpm
ls -l /usr/man/man1/file.1
ls: /usr/man/man1/file.1: 文件或目录不存在
ls -l /usr/man/man4/magic.4
ls: /usr/man/man4/magic.4: 文件或目录不存在
#
9.includedocs：安装说明文档
RPM一般情况下是安装说明文档的，但是如果用户自行修改了设定（象上面的例子），则RPM就不再安装说明文档了，除非特别指定，这就用到--includedocs选项。这种情况下，只有用此选项才能确保安装说明文档。

rpm -i --includedocs file-3.26-6.i386.rpm
ls -l /usr/man/man1/file.1
-rwxr-xr-x 1 root root12023 Mar 23 1999 /usr/man/man1/file.1
ls -l /usr/man/man4/magic.4
-rwxr-xr-x 1 root root 6625 Mar 23 1999 /usr/man/man4/magic.4
#
10.noscripts：不执行脚本程序
一个RPM软件包中可包含五种脚本程序，即：安装前脚本程序，安装后脚本程序，卸载前脚本程序，卸载后脚本程序和校验脚本程序。安装时使用--noscripts选项，可禁止安装前与安装后脚本程序的执行。

rpm -i foo-6.0-1.i386.rpm
preinstall is running ... done
postinstall is running ... done
#
上面不禁止脚本程序的执行，下面不执行脚本程序，请看输出结果：
rpm -i --noscripts foo-6.0-1.i386.rpm
#
看，因为没有安装前与安装后脚本程序的执行，所以没有什么输出了。

注意：一般用户不要使用此选项。此选项主要提供给软件包制作者使用的。通过禁止执行脚本程序，可以防止因安装带有BUG的软件包而宕掉整个系统。当软件包去掉了BUG后，此选项就不必使用了。

11.nodeps：不检查依赖
RPM管理软件包，不仅管理包中的所有文件，还同时管理着软件包之间的依赖关系。如A依赖于B运行，若B不存在了，则A也就运行不了了。RPM维护着这种关系，尽量避免破坏，以保证软件的正常运行。

rpm -i autofs-3.1.3-2.i386.rpm

```
error: failed dependencies:
mktemp is needed by autofs-3.1.3-2
#
```

本例安装中出现了依赖方面的错误（autofs依赖mktemp，但mktemp不存在），安装过程中断了。如果要RPM不管依赖关系是否正常都安装，就要用--nodeps选项，指示RPM不检查依赖，这样就能正常安装了。

```
# rpm -i --nodeps autofs-3.1.3-2.i386.rpm
#
```

注：除非用户对软件包有足够了解，或只想看看软件，否则不要使用本选项，以维护软件间正常的依赖关系。

12.notriggers：不执行触发程序

为了软件包间的交互控制，RPM设计了三种触发程序：安装时触发程序，卸载前触发程序和卸载后触发程序。为了防止因某一软件包的安装而引发安装时触发程序的执行，可以使用--notriggers选项。

13.ignorearch：忽略体系与ignoreos ：忽略操作系统

何谓体系?体系就是CPU的类别，有Intel的x86（如i386，i486系列）体系，有Sun的sparc体系等等。当一个软件包建立时，RPM就为其指定了所适用的CPU体系，也为其指定了所适用的操作系统。这样做的好处就是，RPM容易知道为一台计算机所建立的软件包是否适用于兼容于另一台计算机。RPM的资源配置文件（默认为/usr/lib/rpm/rpmrc）中，就定义了体系的兼容关系（arch_compat表达）和操作系统的兼容关系（os_compat表达）。RPM安装一个包裹文件时，要做这样一个兼容方面的检查。如果用户想忽略体系，不管其是否兼容，请用--ignorearch选项安装。如果也不管操作系统是否兼容，可用--ignoreos选项。需要指出的是，若非知道自己这样的目的，否则不要试图这样做。

14.ignoresize：不检查空间大小

RPM安装软件包前，首先要检查当前系统是否有足够的剩余空间，如果空间不足的话，安装将无法完成。使用本选项的目的，在于指示RPM不做空间大小方面的检查，意即不管系统是否有容纳要安装的软件包的空间，照直安装就行了。

15.relocate：重定位

RPM软件包在制作过程中，可以定义一个或多个重定位前缀，以此方便软件的重定位（即把软件包中的文件放到自定义的目录下面），从而增加软件包安装的灵活性。

本选项用于更换指定的重定位目录，如果一个包有多个重定位前缀，可以使用本选项多次。

```
# rpm -qpl file-3.26-6.i386.rpm
/usr/bin/file
/usr/man/man1/file.1
/usr/man/man4/magic.4
/usr/share/magic
# rpm -i --relocate /usr=/tmp file-3.26-6.i386.rpm
# rpm -ql file
/tmp/bin/file
/tmp/man/man1/file.1
/tmp/man/man4/magic.4
/tmp/share/magic
```

```
#
```

注：本例中先用rpm -qpl列出包裹文件file-3.26-6.i386.rpm当中的文件，可以看到其中的文件均是以/usr开头的。之后进行重定位安装，将/usr换作/tmp。最后从列出已安装的file包的文件列表可以看出，原来的/usr目录前缀换作指定的前缀/tmp了，这样实现了软件的重定位。（关于RPM查询命令，详见<<精通RPM之五--查询篇>>）

16.badreloc：强制重定位

RPM软件包的重定位，依赖于制作时重定位前缀的定义。如果没有定义重定位前缀，或者用户安装时所指定的重定位前缀不存在，则RPM会报错退出，中断安装。此时，若用本选项，则RPM会不管这些错误，进行强制的重定位安装。注：本选项和--relocate选项同时使用时才有意义。

如上例，采用并不存在的重定位前缀/usr/man进行安装：

```
# rpm -i --relocate /usr/man=/tmp file-3.26-6.i386.rpm

path /usr/man is not relocateable for package file-3.26-6
#
```

RPM提示了错误：/usr/man目录在file-3.26-6包中是不可重定位的。

如加上--badreloc选项，结果怎么样：

```
# rpm -i --relocate /usr/man=/tmp --badreloc file-3.26-6.i386.rpm

# rpm -ql file
/usr/bin/file
/tmp/man1/file.1
/tmp/man4/magic.4
/usr/share/magic
#
```

由上可以看到，这样的强制重定位获得通过，之后再用查询命令rpm -ql列一下file包的文件，就会发现原来的/usr/man/man1/file.1定位成了/tmp/man1/file.1，而/usr/man/man4/magic.4则成了/tmp/man4/magic.4。怎么样?神奇吧!

17.excludepath：不安装指定目录下的文件

利用此选项，可以禁止RPM安装某些指定目录下的文件。如不想安装file软件包中的说明文档，除了可使用--excludedocs选项外，还可使用本选项。因为file的说明文档均在目录/usr/man下面。

```
# rpm -i --excludepath /usr/man file-3.26-6.i386.rpm
# rpm -qls file
normal/usr/bin/file
not installed /usr/man/man1/file.1
not installed /usr/man/man4/magic.4
normal/usr/share/magic
#
```

注：例子中先安装file软件包，并用--excludepath禁止安装/usr/man下的文件，再用RPM查询命令rpm -qls列出file包的文件及状态，可以看到：/usr/man下的两个文件均未安装（not installed）。

18.justdb：仅更新数据库

安装软件包使用此选项后，RPM将只更新其数据库数据，文件系统不更新，意即并不拷贝和建立包中的文件。

```
# rpm -i --justdb file-3.26-6.i386.rpm
# rpm -qls file
normal/usr/bin/file
normal/usr/man/man1/file.1
normal/usr/man/man4/magic.4
normal/usr/share/magic
# ls -l /usr/bin/file
ls: /usr/bin/file: 文件或目录不存在
#
```

注：例子中安装file包但指明只更新数据库，结果是：虽然用查询命令查到该软件包中所有文件状态均正常（normal），但用ls命令列包中所含文件/usr/bin/file，该文件却不存在。这表明包中文件并未建立。

19.prefix：指定重定位前缀

如前所述，可重定位的软件包可含一个或多个重定位前缀，可用--relocate选项来改变某个重定位前缀之值。--prefix选项仅能用来改变默认的重定位前缀（默认是第一个重定位前缀），它不需要说明原重定位前缀。下面的例子同样将file包安装到/tmp目录：

```
# rpm -qpl file-3.26-6.i386.rpm
/usr/bin/file
/usr/man/man1/file.1
/usr/man/man4/magic.4
/usr/share/magic
# rpm -i --prefix /tmp file-3.26-6.i386.rpm
# rpm -ql file
/tmp/bin/file
/tmp/man/man1/file.1
/tmp/man/man4/magic.4
/tmp/share/magic
#
```

20.ftpproxy：指定FTP代理主机

本选项指定FTP代理主机，主机可以是主机名，也可以是IP地址。

21.ftpport：指定FTP端口

本选项指定FTP协议使用的TCP端口，系统默认是21。

请看下面的例子：

```
# rpm -iv --ftpport 8888 ftp://root@xwboc/tmp/file-3.26-6.i386.rpm

Password for root@xwboc:
Password for root@xwboc:
Retrieving ftp://root@xwboc/tmp/file-3.26-6.i386.rpm
file-3.26-6
```

#

本例中用FTP方式安装file软件包，数据传输使用8888端口。执行中，RPM先提示输入root用户的密码两次，之后下载（retrieve）软件，最后成功安装。

22.httpproxy：指定HTTP代理主机

本选项指定HTTP代理主机，主机可以是主机名，也可以是IP地址。

23.httpport：指定HTTP端口

本选项指定HTTP协议使用的TCP端口，系统默认是80。下面的例子就指定了80端口来安装autofs软件，当然不指定也行，因为系统默认的就是这个端口。

rpm -iv --httpport 80 http://24.109.164.55/rpms/autofs-3.1.3-2.i386.rpm

Retrieving http://24.109.164.55/rpms/autofs-3.1.3-2.i386.rpm
autofs-3.1.3-2
#
注：同FTP方式一样，RPM都是下载再安装。

二、通用选项

所谓通用选项，就是不管是安装，升级，卸载还是其它软件包操作，统统适用的选项。 这样的选项主要以下几个：

1.-v：显示附加信息

一般情况下，RPM和不少LINUX命令（如cp，rm，mv等）一样，都本着尽可能少输出信息的原则（除非必要，否则不要输出），这样做是为了避免浪费CPU资源。所以，当用户使用某个命令后没有什么输出，这就意味着命令执行成功了。如果有什么错误，程序会提示的。RPM也是这个样子。为了使RPM输出多一点的信息，可以使用-v选项。

如不带此选项安装file包，没有输出：
rpm -i file-3.26-6.i386.rpm
#
而采用-v选项后安装时输出了包名：
rpm -i -v file-3.26-6.i386.rpm
file-3.26-6
#

2.-vv：显示调试信息

选用-vv选项，可以使RPM输出更多的信息。这些信息，主要供RPM软件开发者研究使用的，一般用户也可以看看，从中可以知道RPM究竟干些什么。

下面采用此选项安装file包，同时使用管道线（|）将输出数据送nl命令，以带行号输出每行数据。

从执行结果看，输出的信息很多，其中以D:开头的行均是调试信息。现在逐行解释：
第1行：统计要安装的软件包个数；
第2行：RPM找到1个包；
第3行：查看哪些包需要下载（这些要下载的包，都是以FTP或HTTP形式书写的包名）；
第4行：下载了0个包；
第5-10行：取包裹文件的文件头信息，算出软件包占用的空间68019（字节），以利检查

系统是否有足够空间。
第11行：打开RPM数据库；
第12行：找到0个源代码包，1个执行代码包；
第13-18行：检查软件包的依赖关系。其依赖（requires）的东西均得到满足（satisfied）。
第19行：准备安装执行代码包；
第20行：取得已安装文件系统的列表；
第21-26行：再度取包裹文件头信息，算出软件包占用空间；
第27-31行：列出包中文件及要执行的操作，第27行的test=0表示不是进行安装测试，
file包中的文件均要执行创建操作（create）。
第32行：如果有安装前执行脚本程序的话，执行它；
第33行：安装软件包file-3.26-6；
第34行：如果有安装后执行脚本程序的话，执行它。

3.root：指定根目录

指定根目录ROOT，其作用在于将系统所有操作限定在指定的目录下面，这样RPM操作的数据库位置变了，软件包的安装位置也变了。它是通过chroot系统调用实现的。RPM默认的数据库目录是/var/lib/rpm，如果指定根目录为/usr，则RPM的数据库目录将变为/usr/var/lib/rpm。同样，如果包中有个文件为/etc/zhsoft.txt，则安装后为/usr/etc/zhsoft.txt。

看下面的例子：

```
# rpm -i --root /usr file-3.26-6.i386.rpm
failed to open /usr/var/lib/rpm/packages.rpm
error: 不能打开 /usr/var/lib/rpm/packages.rpm
#
```

注：指定RPM根目录为/usr安装file出现错误，RPM打不开/usr/var/lib/rpm/packages.rpm文件，这是因为实际上在/usr/var/lib/rpm目录下没有任何RPM数据库。如果想成功安装，可用初始化RPM数据库命令来建立RPM数据库，命令是rpm
--initdb --root /usr（这些功能将在以后讲到）。

4.rcfile：指定RPM资源配置文件

RPM的资源配置文件里，存放着RPM的默认设置，有编译选项optflags，体系定义arch_canon，兼容定义arch_compat，宏文件定义macrofiles等。RPM默认的资源配置文件（按读取顺序）为：/usr/lib/rpm/rpmrc，/etc/rpmrc，~/.rpmrc。（后两个文件现在意义不大了，如今一般换作RPM宏文件了。较之资源配置文件，宏文件有更大的灵活性。）

如果想让RPM使用用户指定的资源配置文件，就用--rcfile来设定，可同时设定多个文件，文件间以冒号（:）分隔，如/usr/lib/rpm/rpmrc:/etc/rpmrc:~/.rpmrc这种形式。

5.dbpath：指定RPM数据库目录

本选项不同于--root选项，它仅仅是指定RPM数据库的目录，并不改变安装文件的位置。

2005-1-20 上一页 首页 下一页

10.RPM之认识篇

我们一起在风雨中前行

RPM是RedHat Package Manager的缩写,意即RedHat(红帽子)软件包管理器。(RedHat是美国有名的LINUX公司,网址:http://www.redhat.com
对于一个操作系统来说,不能没有一个象样的软件包管理器。没有软件包管理器的帮助,操作系统发行版的制作者将面临这样或那样的难题,用户安装,升级,卸载与发布软件包也将是非常麻烦的,系统管理也容易出现问题。相反,有了专门的软件包管理器,软件制作者易于制作和发行自己的软件了,而对于普通用户来说,软件包的安装维护将变得非常方便了。这种
情况,对于一个操作系统的推广也会起到良好的促进作用。RPM就是随着RedHat LINUX发行版的流行而迅速推广开来的,二者的表现相得易彰。

RPM先行者

最初的时候,LINUX系统的发布并没有使用什么软件包管理器。随着时间的推移,RedHatmLINUX开发者意识到开发一个软件包管理器的重要性,于是开发出RPP这个管理器。

RPP相对于RPM虽然是简单的,但已有了几项重要的功能,如打一个简单的命令就可以实现软件的安装与卸载,包中可含有安装前后与卸载前后执行的脚本程序,还可以随时校验已安装的软件包,查询功能也很强大。

RPP的缺点在于,RPP打包是基于特别修改过的(针对RPP)源代码的,因而这些源代码并非是纯正的源代码。由于这个原因,当软件包开发者想建立大量不同的软件包时,将面临众多技术面的问题。RPP也不能保证当前的执行程序是基于打包过的源程序的,并且RPP不支持多处理器体系结构。与RPP同时开发的,还有PMS(即package
management system,软件包管理系统),这是另一群LINUX爱好者开发的。PMS采用的是纯正的源代码,它允许软件包制作者很快释出一个软件的最新版本,并且可以立刻看到该软件的变化。RPM采用了这一明智的做法,这也是PMS对RPM的一项重大贡献。PMS的缺点是查询功能不强,没有包校验功能,不支持多体系结构,数据库设计也不好。在RPP和PMS之后,Rik
Faith和Doug Hoffman开发了PM管理器。该管理器整合了RPP与PMS的许多功能,但是数据库设计还不强,依然不支持多体系结构。

RPM开发

此后,Marc Ewing和Erik Troan两人在吸取RPP,PMS,PM设计经验的基础上,用PERL语开发了RPM软件包管理器,即RPM1.0版。

其成功之处在于:

可自动处理配置文件;

可重建大量的软件包;

易于使用。

其不足之处在于:

程序大,运行速度慢,因为它是用PERL这种解释型的语言写的;

数据库功能太弱;

不支持多体系结构;

包裹文件格式不可扩展。

针对RPM1.0的弱点,RPM的开发者再度努力,将RPM升级到2.0,3.0和现在的4.0版本。他们主要做了以下几点:

用C重写了程序,这极大地提高了RPM的运行速度。

RPM数据库格式进行了再设计,重点从性能和可靠性两方面提高。

软件包格式也进行了再设计,方便以后的扩展和升级。

建立了rpmlib(RPM函数库),方便其它程序调用RPM。

增加多体系支持,方便RPM跨平台使用(不论是x86体系,还是sparc等其它体系)。

网上的RPM

RPM官方网址为http://www.rpm.org,

小结

如今的RPM使得软件包安装与卸载更容易,校验已安装的软件包是否正常也容易,将程序(源程序或执行程序)打包也简单了,跨平台的支持,遵循GPL版权发布源代码,使得RPM得到更广泛的应用与推广。RPM正在风靡LINUX与非LINUX世界。如果你想了解甚至精通RPM,那么请跟我来吧!

11.安装RPM包快速应急手册

我们一起在风雨中前行

用RPM包方式来发布软件变得越来越流行，主要的原因是这种软件发布方式很方便用户安装、升级和从系统中删除。除此之外，RPM还可以让你有更多的对系统安装的软件的管理性功能，例如可以查新系统有哪些已经安装的软件包，查看这些软件包分别安装了什么文件，这些文件又放在了什么地方等等。

这里我给出一个简单的入门手册，供初学者查看参考，快速使用RPM。这些RPM命令都是最最常用的，更复杂的功能大家可以暂时不用管，等确实需要的时候再去查man手册。

安装一个RPM包的一般采用的命令格式如下：

```
# rpm -ivh package.ix86.rpm
```

在上面的命令中，package是个假定的名字，ix86是这个包为特定的CPU做了优化处理的意思。例

如，package可能是ldconfig-1.9.5-14，而ix86是i386这样的。

“-i”参数指明是要安装这个package，而“-v”这个参数则使输出信息增加，“-h”表示在安装过程中显示hashes作为在安装过程的一个进度条。

上面就是所有的关于安装一个RPM包的内容，非常简单而好用吧。当然如果你安装过程中遇到了一些包依赖关系的问题那就稍微要麻烦一些了。包依赖关系问题就是说，你正在安装的包需要其他的包的支持才能正常工作。通常还有可能遇到的问题是所安装的包需要某个软件或者函数库的特定版本才能正常工作。这样的情况下，你可以安装或者升级依赖的软件包，然后继续安装你自己需要安装的软件包。

如果你要看一个RPM包安装以后到底安装了那些文件，你可以使用这个命令：

rpm -ql packagename

这个packagename就是你要查询的软件包的名字，例如ldconfig-1.9.5-14mdk；而“-p”参数表示查询软件包数据库，看这个软件包是否已经安装；“-l”参数表示列出所查询的包的所有安装了的文件的完整的路径。

下面是一个例子：

```
[wh@infonet wh]$ rpm -ql ldconfig-1.9.5-14mdk
/sbin/ldconfig
/usr/doc/ldconfig-1.9.5
/usr/doc/ldconfig-1.9.5/README
```

删除某个特定的RPM包软件。采用的命令参数是“-e“，格式如下：

rpm -e packagename

“-e”就表示删除(erase)的意思，就是把特定的软件包（packagename）删除掉。

值得注意的是，rpm管理机制不允许你删除某个软件包，如果这个软件包还被其他软件包所依赖。因为一旦你删除了这个软件包，其他依赖他才能运行的软件包也就不能正常使用了。

升级一个已经存在的rpm包软件。升级其实和安装类似，可以参考下面的命令：

rpm -Uvh package.ix86.rpm

“-U“参数表示是升级（upgrade），参数“v”和“h”意义和前面一样的，这里不再赘述。

到此，我们已经学会了，如何安装、删除和升级一个使用rpm发布的软件了。：）

问题集（FAQ）

目录

FTP的命令行格式

FTP的命令行格式为：
ftp -v -d -i -n -g [主机名] ，其中

-v 显示远程服务器的所有响应信息；

-n 限制ftp的自动登录，即不使用；.n etrc文件；

-d 使用调试方式；

-g 取消全局文件名。

FTP使用的内部命令如下(中括号表示可选项):

1.![cmd[args]]：在本地机中执行交互shell，exit回到ftp环境，如：!ls*.zip

2.$ macro-ame[args]： 执行宏定义macro-name。

3.account[password]： 提供登录远程系统成功后访问系统资源所需的补充口令。

4.append local-file[remote-file]：将本地文件追加到远程系统主机，若未指定远程系统文件名，则使用本地文件名。

5.ascii：使用ascii类型传输方式。
6.bell：每个命令执行完毕后计算机响铃一次。

7.bin：使用二进制文件传输方式。
8.bye：退出ftp会话过程。

9.case：在使用mget时，将远程主机文件名中的大写转为小写字母。

10.cd remote-dir：进入远程主机目录。

11.cdup：进入远程主机目录的父目录。

12.chmod mode file-name：将远程主机文件file-name的存取方式设置为mode，如：chmod 777 a.out。

13.close：中断与远程服务器的ftp会话(与open对应)。

14.cr：使用asscii方式传输文件时，将回车换行转换为回行。

15.delete remote-file：删除远程主机文件。

16.debug[debug-value]：设置调试方式， 显示发送至远程主机的每条命令，如：deb up 3，若设为0，表示取消debug。

17.dir[remote-dir][local-file]：显示远程主机目录，并将结果存入本地文件
18.disconnection：同close。

19.form format：将文件传输方式设置为format，缺省为file方式。

20.get remote-file[local-file]： 将远程主机的文件remote-file传至本地硬盘的local-file。

21.glob：设置mdelete，mget，mput的文件名扩展，缺省时不扩展文件名，同命令行的-g参数。

22.hash：每传输1024字节，显示一个hash符号(#)。

23.help[cmd]：显示ftp内部命令cmd的帮助信息，如：help get。
24.idle[seconds]：将远程服务器的休眠计时器设为[seconds]秒。

25.image：设置二进制传输方式(同binary)。

26.lcd[dir]：将本地工作目录切换至dir。

27.ls[remote-dir][local-file]：显示远程目录remote-dir， 并存入本地文件local-file。

28.macdef macro-name：定义一个宏，遇到macdef下的空行时，宏定义结束。
29.mdelete[remote-file]：删除远程主机文件。
30.mdir remote-files local-file：与dir类似，但可指定多个远程文件，如 ：mdir
.o..zipoutfile 。

31.mget remote-files：传输多个远程文件。

32.mkdir dir-name：在远程主机中建一目录。

33.mls remote-file local-file：同nlist，但可指定多个文件名。
34.mode[modename]：将文件传输方式设置为modename， 缺省为stream方式。
35.modtime file-name：显示远程主机文件的最后修改时间。

36.mput local-file：将多个文件传输至远程主机。

37.newer file-name： 如果远程机中file-name的修改时间比本地硬盘同名文件的时间更近，则重传该文件。
38.nlist[remote-dir][local-file]：显示远程主机目录的文件清单，并存入本地硬盘的local-file。
39.nmap[inpattern outpattern]：设置文件名映射机制， 使得文件传输时，文件中的某些字符相互转换， 如：nmap $1.$2.$3[$1，$2].[$2，$3]，则传输文件a1.a2.a3时，文件名变为a1，a2。该命令特别适用于远程主机为非UNIX机的情况。
40.ntrans[inchars[outchars]]：设置文件名字符的翻译机制，如ntrans1R，则文件名LLL将变为RRR。

41.open host[port]：建立指定ftp服务器连接，可指定连接端口。

42.passive：进入被动传输方式。

43.prompt：设置多个文件传输时的交互提示。

44.proxy ftp-cmd：在次要控制连接中，执行一条ftp命令， 该命令允许连接两个ftp服务器，以在两个服务器间传输文件。第一条ftp命令必须为open，以首先建立两个服务器间的连接。
45.put local-file[remote-file]：将本地文件local-file传送至远程主机。
46.pwd：显示远程主机的当前工作目录。

47.quit：同bye，退出ftp会话。

48.quote arg1，arg2...：将参数逐字发至远程ftp服务器，如：quote syst.
49.recv remote-file[local-file]：同get。

50.reget remote-file[local-file]：类似于get， 但若local-file存在，则从上次传输中断处续传。

51.rhelp[cmd-name]：请求获得远程主机的帮助。

52.rstatus[file-name]：若未指定文件名，则显示远程主机的状态， 否则显示文件状态。

53.rename[from][to]：更改远程主机文件名。

54.reset：清除回答队列。

55.restart marker：从指定的标志marker处，重新开始get或put，如：restart 130。
56.rmdir dir-name：删除远程主机目录。

57.runique：设置文件名只一性存储，若文件存在，则在原文件后加后缀.1， .2等。

58.send local-file[remote-file]：同put。

59.sendport：设置PORT命令的使用。

60.site arg1，arg2...：将参数作为SITE命令逐字发送至远程ftp主机。

61.size file-name：显示远程主机文件大小，如：site idle 7200。

62.status：显示当前ftp状态。

63.struct[struct-name]：将文件传输结构设置为struct-name， 缺省时使用stream结构。

64.sunique：将远程主机文件名存储设置为只一(与runique对应)。

65.system：显示远程主机的操作系统类型。

66.tenex：将文件传输类型设置为TENEX机的所需的类型。

67.tick：设置传输时的字节计数器。

68.trace：设置包跟踪。

69.type[type-name]：设置文件传输类型为type-name，缺省为ascii，如:type binary，设置二进制传输方式。

70.umask[newmask]：将远程服务器的缺省umask设置为newmask，如：umask 3

71.user user-name[password][account]：向远程主机表明自己的身份，需要口令时，必须输入口令，如：user anonymous my@email。
72.verbose：同命令行的-v参数，即设置详尽报告方式，ftp 服务器的所有响 应都将显示给用户，缺省为on.

73.?[cmd]：同help.

上一页 目录 下一页

上一页 目录 下一页

GRUB初级指南

系统引导是非常重要的一部分，很多朋友都碰到了各种各样得问题，这是以前linuxsir上北南南北兄整理得一篇教程，只得借鉴
一。分区：

这也是大家常谈的事了，其实也没有什么难的，但这是最重要的。分区时，最好一次成功。不要给后面安装操作系统带来麻烦。比如空间的大小，分区的个数等。主、扩展以及逻辑分区的个数。这些都是最重要的。

1.硬盘划分为什么很重要？

您可以用dos下的PQ，可能这是最简单的分区工具了。主分区最好只有一个，扩展分区也是一个了。逻辑分区那就根据自己的需要来分。分区要规划好，以防为以后安装操作系统带来麻烦。要有个预期，也就是说，以防您以后安装操作系统，而没有空间。所以我们就要划分硬盘要好好的规划。以满足我们未来不可预期情况的需要。

2.我应该把硬盘划分为多少个区，每个区应该有多大？

对于这个问题，可能有的弟兄很是疑惑。其实对于我们玩家来说，无非是学习和实践。所以我们要为每

个linux划分为一个区，空间大小为6G以上。现在的linux体积都很大，如果是初次玩linux，我还是建议完全安装，这样便于您的学习，否则少东少西的，自己都不知道为什么。等玩的技术差不多时，你再根据自己的需求来选择安装组件。

举个例子，比如我想在一块硬盘上安装两个linux和一个windows，那我就可以这样来划分硬盘。把windows的分区放在硬盘的第一个分区。这个分区大家常用来安装windows系统。如果您还想为windows划分应用程序的分区以及备份文件的分区，那就把第二个和第三个等分区做为windows的分区。linux分区要放在windows的分区的后面，应该避免把windows和linux的分区交叉。我们可以用PQ把linux分区分好。为每个linux分一个大一点的分区6G以上，这里，我们将要在每个分区里，要安装一个linux，其实每个6G左右的分区，就是我们要安装linux的根分区。然后再划分一个小一点的分区，大小为内存的两倍就行。如果用PQ来划分。linux的分区格式，可以不用指定。因为安装系统时，也会有相关的步骤来格式化磁盘。

仔细看一下下面的例子：
在这里，我把安装了，两个linux和一个windows，分区情况如下。主分区是hda1，格式为ntfs，系统是XP，hda2是扩展分区，hda5是逻辑分区，格式为nfts，是windows的安装应用软件的磁盘,hda6,hda7是逻辑分区，格式为fat32的，也是windwows用的，我用来备份和存放常用程序和文档的磁盘。hda8和hda9是linux的磁盘。看一下应该能看得出来，大小为6-7G左右吧。在hda8和hda9，我每个区都安装了一个linux，也就是说，这两个分区是不同的linux的/根分区。hda10也是linux的磁盘，大小是内存的1.5倍，如果您的内存为512M或者以上的，交换分区和内存大小一样就行，格式为/swap。也就是linux的交换分区。

linux的分区，没有windows下面的C盘和D盘的概念。这一点大家应该明白。

一块磁盘最多只能分4个主分区。扩展分区也算一个主分区，所以为了简便 ，我们一块硬盘就分一个主分区，一个扩展分区，其它的都是逻辑分区。下面的例子就是。这样做的好处就是在安装linux的时候，分区表不会错，也不会出现这样或者那样危机windows分区的情况。

在linux中，逻辑分区是从hda5开始的，当然这是说一台机器中只有一个硬盘的情况来说的。如果有两个硬盘，可能就是从hdb5开始，以此类推。如果您不知道，是怎么回事，就要用fdisk -l来查看。要安装系统的时候，也会有相应的让您选择安装位置 ，也是hdaX或者 hdbX分区之类的，没有WIN的C和D盘的概念。

```
[root@linuxsir01 root]# fdisk -l

Disk /dev/hda: 255 heads, 63 sectors, 4865 cylinders
Units = cylinders of 16065 * 512 bytes

Device Boot Start End Blocks Id System
/dev/hda1 * 1 1147 9213246 7 HPFS/NTFS
/dev/hda2 1148 4864 29856802+ f Win95 Ext'd (LBA)
/dev/hda5 1148 2039 7164958+ 7 HPFS/NTFS
/dev/hda6 2040 2804 6144831 b Win95 FAT32
/dev/hda7 2805 3824 8193118+ b Win95 FAT32
/dev/hda8 3825 3830 7768163+ 83 Linux
/dev/hda9 3831 4799 7783461 83 Linux
/dev/hda10 4800 4864 522081 82 Linux swap
```

3.硬盘的分区划好后，我如何安装linux呢？

我们可以先安装windows，然后再安装linux，这样做的目的就是用grub能把windows引导起来。如果先安装linux，可能windows就把grub从MBR中删除。我们如果重装windwows经常会出现这样的情况，方法只是把GRUB恢复一下就行了。所以为了方便，我们还是先安装windows。这样就减少不少劳动。呵。。。。。

接着举例：

在这里，我是先安装Mandrake 9.0，到hda8，这样做的，在安装过程中，有个[自定义]，在划分硬盘和安装软件时都有[自定义]。我们就选这一项，然后把光标移动到hda8,[编辑，指定分区格式和挂载点。格式为ext3，挂载点为/。OK一下，然后再移动hda10，我们编辑这个分区，格式为/swap。这样就简单了。分区都处理好了，就是选择组件来来装。下面就是点鼠标。应该简单。。。一直到最后。

然后我们安装redhat 8.0到hda9，也就是说，我们要把redhat 8.0的根分区设置为hda9。安装过程也一样的。。在划分磁盘是，要[自定义]用disk druid 来分区。光标移动到hda9，然后[编辑]，格式为ext3，挂载点为/，可能会跳出一个对话框，那就anyway.吧。不用再编辑交换区了，因为系统能认得出hda10是交换区，不用我们再设置了。linux的交换区是能共用的。我们把hda9设置好后，确认一下，就进入软件安装的过程了。也是[自定久]。。。下面的就一路下去，不用我多说了。

到GRUB的设置的那一块。我们要把GRUB安装到MBR上。这样就把mandrake的GRUB覆盖。这也没有什么可怕的，我们可以把它加在Redhat 的GRUB中，系统安装好后，我们设置一下就行了。只是修改一个文件/etc/grub.conf

4.如何修改/etc/grub.conf
一般的发行版，GRUB的配置文件都在/etc/grub.conf中，如果我们安装了多个linux，就要修改这个文件。那我们如何修改呢？一般的情况下，windows是自动加载到了grub.conf中，打开grub.conf，就能看到windows的设置情况。可能有时系统会突然出现不能引志windwows的情况，我们就要下面的类似的改动一下。
title DOS
rootnoverify (hd0,0)
chainloader +1

改成
title WinXP在这里,title是必须的，WinXP，您可以用别的名，您喜欢怎么样就怎么样。
root (hd0,0)
chainloader +1

这样就能启动了。

2.如何把其它的linux的引导加入到GRUB来引导呢？我们可以用下面的方法。也是比较简单。GRUB引导linux系统，就是靠三行来定义。

title XXX
在这里，XXX是您要引导的操作系统的名字。可以自己来定。

root (hdX,Y)
在这里，X和Y都代表一个数值。X是linux的根分区在哪个硬盘上。如果您只有一个硬盘。那就是hd0。如果您的linux的一个发行版是安装在第二个硬盘，这个X，就是1。那Y代表什么？Y是代表您所安装的linux是在哪个区。这个数值有点不同。用原理没有办法说清楚。应该用实例来说。比如我把mandrake安装到hda8了。那他的root (hdX,Y)，就应该写为root (hd0,7)，也是说是linux所在的根分区hdaZ中的Z减去1就是这个Y的数值：在这之中，hdaZ或者 hdbZ中的Z也是一个数值。看一下上面的例子就明白了。用fdisk -l 或者df -la来查看

kernel /boot/vmlinuz root=/dev/hdaZ
这里中的Z，也是一个数值，接着上面的例子，我们把Mandrake的引导加入到Redhat 的GRUB中。就可以写成如下的样子：

kernel /boot/vmlinuz root=/dev/hda8

通过上面的分析，我们要让Redhat的GRUB来引导Mandrake 9.0，我们要改动Redhat中的GRUB的配置文件，加入下面的一段：

title MD 8.0
root (hd0,7)
kernel /boot/vmlinuz root=/dev/hda8

如果您想加入其它的操作系统到Redhat 中的GRUB中，也不过是如此，如者用其它发行版中的GRUB来引导别的发行版，原理也是一样的。

上一页 目录 下一页

上一页 目录 下一页

GRUB三步通

朋友，你装好linux后是不是每次启动后系统就默认进入到linux，想让它变成windows吗？请修改你的引导装入器吧（当然它不止这个功能）！由于现在的linux都喜欢用GRUB来引导，LILO已越来越少用了，所以我在这里主要给大家讲讲GRUB这个东东！！

##################
GRUB的优点
##################
GRUB 是引导装入器(boot loader) -- 它负责装入内核并引导 Linux 系统。GRUB 还可以引导其它操作系统，如 FreeBSD、NetBSD、OpenBSD、GNU HURD 和 DOS，以及 Windows 95、98、NT 和 2000。尽管引

导操作系统看上去是件平凡且琐碎的任务，但它实际上很重要。如果引导装入器不能很好地完成工作或者不具有弹性，那么就可能锁住系统，而无法引导计算机。另外，好的引导装入器可以给您灵活性，让您可以在计算机上安装多个操作系统，而不必处理不必要的麻烦。
GRUB 是一个很棒的boot loader。它有许多功能，可以使引导过程变得非常可靠。例如，它可以直接从 FAT、minix、FFS、ext2 或 ReiserFS 分区读取 Linux 内核。这就意味着无论怎样它总能找到内核。另外，GRUB 有一个特殊的交互式控制台方式，可以让您手工装入内核并选择引导分区。这个功能是无价的：假设 GRUB 菜单配置不正确，但仍可以引导系统。哦，对了 -- GRUB 还有一个彩色引导菜单。

更令人惊讶的是，这是一个自由软件！！！

```
##################
# GRUB菜单 #
##################
```

先来看一个例子，这是位于/boot/grub/目录下的menu.lst文件。
此文件将在开机是产生一个菜单，包含有Debian linux,Windows2000,RedHat linux和 Mandrake linux,共四个选择项。我一共分了8个区，一个fat16（0x6），一个ntfs（0x7），三个ext2fs分区（0x83），一个swap分区（0x82）。ntfs用来装win2000，三个ext2fs装了三个linux，c盘fat16分区没有装任何东西。
＃例子由此开始

```
＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃
#  #
#  一个GRUB configure 的例子 #
#  #
＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃＃

timeout 10
default 2


# --> Debian linux <--

title Debian linux
root (hd0,2)
kernel /boot/vmlinuz-2.2.18 root=/dev/hda3 ro
initrd /boot/initrd-2.2.18.gz


# --> Debian END <--

# --> Windows 菜单选项 <--

title Windows2000
root (hd0,0)
chainloader +1
```

```
# --> Winddows 结束 <--

# --> RedHat linux 菜单选项 <--

title RedHat linux
root (hd0,8)
chainloader +1 # 在硬盘主引导分区装了lilo，所以也用了chainloader。

# --> RedHat linux 结束 <--

# --> Mandrake linux 菜单选项 <--

title Mandrake linux
root (hd0,5)
kernel /boot/vmlinuz-2.4.3-20mdk root=/dev/hda6 ro
initrd /boot/initrd-2.4.3-20mdk.img

# --> Mandrake linux 结束 <--
```

＃例子到此结束

以符号井"＃"开头的行表示被注释掉，没有任何意义。

timeout表示默认等待的时间，这儿是10秒钟。超过10秒，用户还没有作出选择的话，系统将自动选择默认的操作系统。

默认的操作系统就是由default控制的。default后加一个数字n，表明是第n＋1个。需要注意的是，GRUB中，计数是从0开始的，第一个硬盘是hd0，第一个软驱是fd0，等等。所以，default 2 表示默认的操作系统在这儿是 Redhat linux。

接下来，正如你所想象的，title表示的是“Debian linux”菜单项。root (hd0,2)表示第一个硬盘,第三个分区。这儿的root 于linux的root分区及其不同，此root非彼root也！

在 Linux 中，当谈到 "root" 文件系统时，通常是指主 Linux 分区。但是，GRUB 有它自己的 root 分区定义。GRUB 的 root 分区是保存 Linux 内核的分区。这可能是您的正式 root 文件系统，也可能不是。我们讨论的是 GRUB，需要指定 GRUB 的 root 分区。进入 root 分区时，GRUB 将把这个分区安装成只读型，这样就可以从该分区中装入 Linux 内核。GRUB 的一个很“酷”的功能是它可以读取本机的FAT、FFS、minix、ext2 和 ReiserFS 分区。

到目前为止，您可能会感到一点疑惑，因为 GRUB 所使用的硬盘／分区命名约定与 Linux 使用的命名约定不同。在 Linux 中，第一个硬盘的第五个分区称作 "hda5"。而 GRUB 把这个分区称作 "(hd0,4)"。GRUB 对硬盘和分区的编号都是从 0 开始计算。另外，硬盘和分区都用逗号分隔，整个表达式用括号括起。现在，可以发现如果要引导 Linux 硬盘 hda5，应输入 "root (hd0,4)"。

知道了内核在哪儿，还要具体指出哪个文件是内核文件，这就是kernel的工作。

kernel /boot/vmlinuz-2.2.18 root=/dev/hda3 ro说明/boot/vmlinuz-2.2.18 就是要载入的内核。后面的都是传递给内核的参数。root=/dev/hda3就是linux的硬盘分区表示法，ro是以readonly的意思。
initrd用来初始的linux image，并设置相应的参数。

是不是感觉很简单啊！再来看一看windows的定义段吧。
这里，我添加了一项来引导 Windows2000。要完成此操作，GRUB 使用了“链式装入器”(chainloader)。链式装入器从分区 (hd0,0) 的引导记录中装入 win2000 自己的引导装入器，然后引导它。这就是这种技术叫做链式装入的原因 -- 它创建了一个从引导装入器到另一个的链。这种链式装入技术可以用于引导任何版本的 DOS 或 Windows。

我的RedHat linux在硬盘主引导分区装了lilo，所以也用了chainloader。

GRUB的配置文件要简单就这么简单，如果你要更个性化一点，试一试把“color light-gray/blue ”加在default语句的下面，下一次启动GRUB时，看看有什么变化，再试一试“color light-blue/red",惊喜吗？ 有趣吧!

######################
GRUB的交互性
######################

GRUB 最好的优点之一就是其强健的设计 -- 在不断使用它时请别忘了这点。如果更新内核或更改它在磁盘上的位置，不必重新安装 GRUB。事实上，如有必要，只要更新 menu.lst 文件即可，一切将保持正常。

只有少数情况下，才需要将 GRUB 引导装入器重新安装到引导记录。首先，如果更改 GRUB root 分区的分区类型（例如，从 ext2 改成 ReiserFS），则需要重新安装。或者，如果更新 /boot/grub 中的 stage1 和 stage2 文件，由于它们来自更新版本的 GRUB，很有可能要重新安装引导装入器。其它情况下，可以不必理睬!

GRUB的最大的特点就是交互性特别强。在开机时，按一下“c”，将进入GRUB 控制台。显示如下：

GRUB version 0.5.96.1 (640K lower / 3072K upper memory)

[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]

grub>

欢迎使用 GRUB 控制台。现在，再研究命令：
我将通过GRUB 控制台绕过lilo来启动RedHat linux，

grub> root (h

现在，按一次 Tab 键。如果系统中有多个硬盘，GRUB 将显示可能完成的列表，从 "hd0" 开始。如果只有一个硬盘，GRUB 将插入 "hd0,"。如果有多个硬盘，继续进行，在 ("hd2") 中输入名称并在名称后紧跟着输入逗号，但不要按 Enter 键。部分完成的 root 命令看起来如下：

```
grub> root (hd0,
```

现在，继续操作，再按一次 Tab 键。GRUB 将显示特定硬盘上所有分区的列表，以及它们的文件系统类型。在我的系统中，按 Tab 键时得到以下列表：

```
grub> root (hd0, (tab，按tab一下键)
Possible partitions are:
Partition num: 0, Filesystem type is fat, partition type 0x6
Partition num: 2, Filesystem type is ext2fs, partition type 0x83
Partition num: 4, Filesystem type unknown, partition type 0x7
Partition num: 5, Filesystem type is ext2fs, partition type 0x83
Partition num: 6, Filesystem type is fat, partition type 0xb
Partition num: 7, Filesystem type is fat, partition type 0xb
Partition num: 8, Filesystem type is ext2fs, partition type 0x83
Partition num: 9, Filesystem type unknown, partition type 0x82
```

如您所见，GRUB 的交互式硬盘和分区名称实现功能非常有条理。这些，只需要好好理解 GRUB 新奇的硬盘和分区命名语法，然后就可以继续操作了

```
grub> root (hd0,8)
```

现在已安装了 root 文件系统，到装入内核的时候了

```
grub> kernel /boot/vmlinuz-2.4.2 root=/dev/hda5 ro
[Linux-bzImage, setup=0x1200, size=0xe1a30]
```

您已经安装了 root 文件系统并装入了内核。现在，可以引导了。只要输入 "boot"，Linux 引导过程就将开始。是不是很cool啊，GRUB的menu.lst更像一个linux下的脚本程序。

#####################
GRUB启动盘
#####################

要制作引导盘，需执行一些简单的步骤。首先，在新的软盘上创建 ext2 文件系统。然后，将其安装，并将一些 GRUB 文件复制到该文件系统，最后运行 "grub" 程序，它将负责设置软盘的引导扇区。准备好了吗？

将一张空盘插入 1.44MB 软驱，输入：

```
# mke2fs /dev/fd0
```

创建了 ext2 文件系统后，需要安装该文件系统：

```
# mount /dev/fd0 /mnt/floppy
```

现在，需要创建一些目录，并将一些关键文件（原先安装 GRUB 时已安装了这些文件）复制到软盘：

```
# mkdir /mnt/floppy/boot
# mkdir /mnt/floppy/boot/grub
# cp /boot/grub/stage1 /mnt/floppy/boot/grub
# cp /boot/grub/stage2 /mnt/floppy/boot/grub
```

再有一个步骤，就能得到可用的引导盘。

在linux bash中，从 root 用户运行“grub”，该程序非常有趣并值得注意，因为它实际上是 GRUB 引导装入器的半功能性版本。尽管 Linux 已经启动并正在运行，您仍可以运行 GRUB 并执行某些任务，而且其界面与使用 GRUB 引导盘或将 GRUB 安装到硬盘 MBR 时看到的界面(即GRUB控制台)完全相同。
在 grub> 提示符处，输入：

```
grub> root (fd0)
grub> setup (fd0)
grub> quit
```

现在，引导盘完成了。
如果要把GRUB装到硬盘上，也很容易。这个过程几乎与引导盘安装过程一样。首先，需要决定哪个硬盘分区将成为 root GRUB 分区。在这个分区上，创建 /boot/grub 目录，并将 stage1 和 stage2 文件复制到该目录中，可以通过重新引导系统并使用引导盘，或者使用驻留版本的 GRUB 来执行后一步操作。在这两种情况下，启动 GRUB，并用 root 命令指定 root 分区。例如，如果将 stage1 和 stage2 文件复制到 hda5 的 /boot/grub 目录中，应输入 "root (hd0,4)"。接着，决定在哪里安装 GRUB -- 在硬盘的 MBR，或者如果与 GRUB 一起使用另一个“主”引导装入器，则安装在特定分区的引导记录中。如果安装到 MBR，则可以指定整个磁盘而不必指定分区，如下（对于 hda）：

```
grub> setup (hd0)
```

如果要将 GRUB 安装到 /dev/hda5 的引导记录中，应输入：

```
grub> setup (hd0,4)
```

现在，已安装 GRUB。引导系统时，应该立即以 GRUB 的控制台方式结束（如果安装到 MBR）。现在，应创建引导菜单，这样就不必在每次引导系统时都输入那些命令。

小结：在这里只是介绍了 GRUB 的一部分。例如，可以使用 GRUB 来执行网络引导，引导 BSD 文件系统，或更多操作。另外，GRUB 有许多配置和安全性命令也很有用。如需所有 GRUB 功能的完整描述，请阅读 GRUB 出色的 GNU 文档。只要在 bash 提示中输入 "info grub" 就可以阅读该文档。

上一页 目录 下一页

上一页 目录 下一页

LINUX 常见问题1000个详细解答

一、 如何建立多用户

提醒大家一句，别一直使用root用户，因为root用户在系统中有着至高无上的权力，一不小心
就可能破坏系统。比如我们想删除/temp目录下的文件却将命令不小心输成“rm / temp（在‘/’后
多了一个空格）”，那么就极可能删去根目录下的所有文件。再者，“Linux是一个真正

意义上的多
任务、多用户系统”，不体会一下Linux的特色岂不可惜。为了方便自己和别人使用，就

在Linux系统
下多建几个用户吧。

输入“adduser newuser”，回车，屏幕对命令没有一点儿反应，错啦？不，实际上

，系统已接受
了新用户。
你应该接着输入命令 “passwd newuser”，屏幕将提示输入口令，再次输入确认， 一个新的用户就增
加成功，
真简单。马上登录试一试吧。

Linux提供了六个虚拟控制台，用ALT+F1~F6切换（在X-window下用CRTL+ALT+F1~F6切换），
每个控制台均可
以用不同的用户登录，运行不同的程序。赶快为家人和朋友建个用户吧。

二、在Linux下如何使用软盘、光盘以及DOS等非Linux分区

这是Linux初学者问得最多的问题。由于大家已习惯了微软的访问方法，总想用类似

的思路来找
到软盘和光盘。
但在Linux下，却沿袭了UNIX将设备当作文件来处理的方法。所以要访问软盘和光盘，就

必须先将
它们装载
到Linux系统的/mnt目录中来。

装载的命令是mount，格式如下：mount -t 文件系统类型 设备名 装载目录

文件系统类型就是分区格式，Linux支持的文件系统类型有许多：

msdos DOS分区文件系统类型

vfat 支持长文件名的DOS分区文件（可以理解为Windows文件）系统类型

iso9660 光盘的文件系统类型

ext2 Linux的文件系统类型

......

设备名，指的是你要装载的设备的名称。软盘一般为/dev/fd0 fd1；光盘则根据你的光驱的位置
来决定，
通常光驱装在第二硬盘的主盘位置就是/dev/hdc;如果访问的是DOS的分区，则列出其设备名，

如/dev/hda1是指第一硬盘的第一个分区。 装载目录，就是你指定设备的载入点。

1. 装载软盘

首先用mkdir /mnt/floppy 在/mnt目录下建立一个空的floppy目录，然后输入:
mount -t msdos /dev/fd0 /mnt/floppy 将DOS文件格式的一张软盘装载进来，以后就可

以
在/mnt/floppy目录
下找到这张软盘的所有内容。

2. 装载Windows所在的C盘

mkdir /mnt/c 在/mnt目录下建立一个空的c目录;

mount -t vfat /dev/hda1 /mnt/c 将Windows的C盘按长文件名格式装载到/mnt/c目

录下，
以后在该目录下
就能读写C盘根目录中的内容。

3.装载光盘

mkdir /mnt/cdrom 在/mnt目录下建立一个空的cdrom目录;

mount -t iso9660 /dev/hdc /mnt/cdrom 将光盘载入到文件系统中来，将在/mnt/
cdrom目录下
找到光盘内
容。有的Linux版本允许用mount /dev/cdrom或mount /mnt/cdrom命令装载光盘。

要注意的是，用mount命令装入的是软盘、光盘，而不是软驱、光驱。有些初学者容

易犯一
个毛病，以为用
上面命令后，软驱就成了/mnt/floppy，光驱就成了/mnt/cdrom，其实不然，当你要换一

张光盘或软
盘时，一定
要先卸载，再对新盘重新装载。

4.卸载

卸载的命令格式是:umonut 目录名，例如要卸载软盘，可输入命令umonut /mnt/fl
oppy。要注意的
是，在
卸载光盘之前，直接按光驱面板上的弹出键是不会起作用的。

三、如何安装Linux的应用软件

对于普通用户，安装或升级软件可以用Red Hat公司的软件包安装与管理工具RPM。
使用后你就可以轻松安装
软件、升级软件和卸载软件并查询软件信息。

如果你安装的是Red Hat Linux，那么你就可以使用RPM了，如果不是，你可以先安
装RPM。方法如下：

1. 下载RPM软件包：ftp://ftp.redhat.com/pub/code/rpm/rpm-2.3.11.tar.gz
?
LINUX 常见问题1000个详细解答
一、 如何建立多用户

提醒大家一句，别一直使用root用户，因为root用户在系统中有着至高无上的权力
，一不小心就可能破坏系统。

系列教材二

目录

GRUB三步通

我们一起在风雨中前行

GRUB三步通
###################
GRUB的优点
###################
GRUB 是引导装入器(boot loader) -- 它负责装入内核并引导 Linux 系统。GRUB
还可以引导其它操作系统，如 FreeBSD、NetBSD、OpenBSD、GNU HURD 和 DOS，以及 Windows 95、98、NT
和 2000。尽管引导操作系统看上去是件平凡且琐碎的任务，但它实际上很重要。如果引导装入器不能很好地完成工作或者不具有弹性，那么就可能锁住系统，而无法引导计算机。另外，好的引导装入器可以给您灵活性，让您可以在计算机上安装多个操作系统，而不必处理不必要的麻烦。

GRUB 是一个很棒的boot loader。它有许多功能，可以使引导过程变得非常可靠。例如，它可以直接从 FAT、minix、FFS、ext2
或 ReiserFS 分区读取 Linux 内核。这就意味着无论怎样它总能找到内核。另外，GRUB 有一个特殊的交互式控制台方式，可以让您手工装入内核并选择引导分区。这个功能是无价的：假设
GRUB 菜单配置不正确，但仍可以引导系统。哦，对了 -- GRUB 还有一个彩色引导菜单。更令人惊讶的是，这是一个自由软件！
###################
GRUB菜单
###################
先来看一个例子，这是位于/boot/grub/目录下的menu.lst文件。
此文件将在开机是产生一个菜单，包含有Debian linux,Windows2000,RedHat linux和 Mandrake linux,

备有与Linux沟通的工具，
怎么办呢？别急，网络上有个小工具叫Linuxdos，不到500K，运行在Windows下，专用于

查看Linux的文件。运行
之后，你可以选择驱动器，然后它就像资源管理器一样列出Linux的文件系统。
五、如何汉化Linux
　　Linux的母语是英文，所以满屏幕都是“英国老鼠”，用起来特不方便，如果想输入

中文更没辙。不过现在
的中国软件人员可谓人才辈出，已经为Linux的汉化提供了全面的解决方案。
　　1.中文模拟终端chdrv
　　首先，总得让Linux能显示中文吧！chdrv是在命令行下的中文模拟终端，也就是说
，只要你装上了它，你就
可以在Linux的命令行状态下显示和输入汉字了。chdrv启动后在tty7（用Ctrl+Alt+F7）

建立一个虚拟终端，用户
在tty7上登录后就可以使用它提供的中文支持。tty1到tty6不会受到它的影响。
　　如果你能找到RPM格式的安装包，就可以轻松地使用RPM安装程序安装它。否则你可
以
到ftp://ftp.ihep.ac.cn/pub/chinese/systemchdrv 的已编译的chdrvbin-0
.23.gb.tar.gz和
字体文件chdrvfont.tar.gz。
然后用root登录，依次执行以下命令:
　　tar xvzf chdrvbin-0.23.gb.tar.gz 将编译好的chdrv解压到chdrv-0.23.gb目录
　　mv chdrvfont.tar.gz chdrv-0.23.gb/ 将字体文件移到chdrv-0.23.gb目录
　　cd chdrv-0.23.gb 进入此目录
　　./installbin 运行安装文件
　　2.中文输入服务器xcin
　　如果你常在X-Window中使用Linux，一定试试用XCIN来解决中文输入的问题。XCIN
是Xwindow Chinese INput的
缩写，是一个在X-Window模式下运行的中文输入系统，安装它后，X-Window的所有终端
都将提供中文服务。
　　你可从ftp://ftp.ihep.ac.cn/pub/chinese/system/xcingb-2.2.tar.gz
的XCIN文件包。然后
用root登录，依次执行以下命令：
　　tar xzvf xcingb-2.2.tar.gz　　解压缩
　　cd xcingb-2.2　　　　　　　　　进入目录
　　./configure　　　　　　　　　　修改安装选项
　　make　　　　　　　　　　　　　编译
　　make install　　　　　　　　　　安装
　　同样，如果你能找到该软件的RPM安装版，可以用RPM来安装。
　　到此为止，我们已经介绍了在命令行下和X-Window中使用中文的方法。可是在文本
编辑软件中依然只能显
示中文，却无法输入汉字。要解决这个问题必须自己动手修改两个地方:
　　1） 为不同shell设置
　　若使用Bash Shell:在 /etc/profile 增加下面的内容:
　　stty cs8 -istrip

stty pass8

export LANG=C

export LC_CTYPE=iso-8859-1

若使用Tcsh Shell:在 /etc/csh.login 或 /etc/csh.cshrc 增加设定如下:

stty cs8 -istrip

stty pass8

setenv LANG C

setenv LC_CTYPE iso-8859-1

2） 在自己主目录下的.inputrc文件中增加下面的内容：

set convert-meta off

set output-meta on

3.中文的窗口管理器 C-fvwm95

fvwm95就像是Linux英文Win95，中国的Linux用户一直期望有一天能在中文的窗口下

使用Linux。这一愿望
现在终于成为现实。中国科学院高能物理研究所计算中心的于明俭先生将fvwm95作了汉化，我们终于可以
将fvwm95上那些讨厌的英文换成汉字了！

大家可以到ftp.ihep.ac.cn/pub/chinese/packages/中下载汉化了的fvwm95的源代码。它所用的字库也可
以在ftp.ihep.ac.cn/pub/chinese/fonts/pcf/中取得。下载文件中附带了安装说明。由

于它的配置还是较麻烦
的，所以最好去找找相应的RPM包。

4.下载BluePoint Linux Star Preview（蓝点预览版）

蓝点预览版基于Red Hat，内含：中文内核(kernel)；控制台中文显示/输入；中文

Xserver(CXWIN 0.6)，X的
中文显示/输入；汉化Message和相应库文件，经过中文化和本地化处理的NLS message和

库文件。

a. 中文内核

1. 性能卓越, 内核实现中文支持，占用资源极少，速度极快.

2. 多语言支持， 同屏显示多内码(GB/BIG5).

3. 兼容性极好，内核兼容上层应用程序，最彻底的解决办法.

4. 支持高分辨率, 完全取决与用户自己的硬件限制.

5. 支持各种字体，支持简繁字体共14种.

6. 热键切换语言环境(GB/BIG5/EN).

b. 中文Xserver

1. 兼容性好，至少95%的纯西文应用软件可以支持中文处理.

2. 内置输入和显示支持，使用非常方便,无须另外安装输入模块.

3. TTF字库支持.

4. 支持GB/BIG5内码.

c. 汉化Message和相应库文件

1. 提示中文化，开机、关机和常用软件的消息提示全部中文显示。

如果需要安装中文xwindow和输入法,则需要下载star-cxwin.tar.gz. 如果需要安装中文控制台及语言环

境, 则需要下载star-zhcon.tar.gz 如果需要在控制台上使用24点阵汉字库, 请下载star-font24.tar.gz 注
意: 别忘了下载Install安装脚本! 在安装前请仔细阅读README和 常见问题问答!

安装方法

1) 创建一个目录, 将所有下载来的文件放在里面.

2) 以root身份执行安装脚本: # bash Install

3) 安装脚本会先检查系统的配置,还会自动搜索TTF字库, 接下来就是对旧配置文件 进行备份, 最后
开始安装.

4) 完成安装, 如果安装了中文控制台环境, 则需要重新启动.

六、如何设置声卡

不用我说，这可是十分要紧的工作！不过如果你安装了RED HAT Linux，则这项工作

就简单了！

用root登录系统，运行/usr/sbin/sndconfig。出现选择声卡类型的画面。一般声卡

可以选取Sound Blaster 。

选完声卡后再设置有关的资源。用TAB键结合方向键作出选择后，按OK，如果听到Linus（Linux的创始人）
说话的声音，就说明声卡设置成功了。真像老式DOS游戏的声卡设置……

七、如何设置显卡

要设置好你的显卡，首先，要知道你的显卡的型号，是什么公司出的，什么样的类

型的显卡，显存有
几MB，还要知道显卡的显示芯片是什么类型的，是ALG2302的，还是SIS6326的，
还是Savage3D的，等；然后，在超级用户的权限下，你可以运行界面比较友好的Xconfigurator工具，
这里的X记住一定要大写的，一步一步跟着指导来选择，应该不是很难的，当选择你的显

卡的芯片的时钟时，
不用选择它，让测试程序自动检查；最后，
还要知道你的显示器的类型，是VGA的，还是SVGA的，以及水平和垂直分辩率。你还可以

运行界面不友好的xfree86，
如果你比较精通Linux的话，用xfree86命令配置的X Windows效果比Xconfigurator好很多的。界面最友好的，当
属XF86Setup工具，若你有安装这个工具的话，不妨就用这个工具来设置你的显卡吧。

八、如何设置网卡

九、如何播放CD音乐

声卡设置好了，可以在光驱中插入一张CD，用装载命令mount将光盘载入，然后输入

cdp命令就可以播放了。
在屏幕上可以看到CD的音轨清单，小键盘的9是播放，7是停止，6进到下一音轨，4退到前一音轨，0退出播放程序，
2弹出CD，8是暂停。

如果是在X-Window中，可以用xplaycd播放CD，这是一个有图形界面的播放器，按钮

及功能一目了然，这里
就不多说了。
10、如何播放VCD影碟
11、如何拨号上网
12、Linux下能玩游戏吗
　　光盘中有bsd-games-2.1-3.src.rpm文件包，用RPM命令进行安装，然后到安装目录中去找游戏吧。不过
这些游戏都不够精致，如果你想玩好的，就须要进入X-Window，acm是空战模拟游戏，paradise和xpilot是联
网战斗游戏，xdemineur是挖地雷（没想到吧），xjewel是俄罗斯方块，xboard是国际象

棋，xpat2是扑克牌游
戏，xboing是弹珠台游戏，还有Doom——大名鼎鼎的第一人称射击游戏！这些游戏有的可以直接调出，有的
须要用RPM命令安装。所有的RPM包都在安装光盘中的srpms目录下，自己去看看吧。
十.用xvidtune调整你的显示器
大家会发现装了linux之后在windows下用的好好的显示器有时进到linux的xwindows里后

就歪掉了，调整好之
后回到windows后windows的桌面也外调了，来回启动系统每次都要调整很麻烦的，这里介绍一个办法一劳永逸
j进入linux启动x在xterm里执行xvidtune,会弹出这个软件的窗口，点Auto然后点Left,Right等按钮调整你的
显示器到最佳的位置，然后点界面上的Show按钮会得到类似这样的输出：
"1152x864" 121.50 1152 1232 1360 1568 864 865 868 911 +hsync -vsync
然后退出这个软件，修改你的/etc/X11/XF86Config-4文件在
Section "Monitor"
Identifier "AS 786T"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 30 - 87
VertRefresh 50 - 160
Option "dpms"
EndSection
里加上刚才的输出，我的是：
Section "Monitor"
Identifier "AS 786T"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 30 - 87
VertRefresh 50 - 160
Modeline "1152x864" 121.50 1152 1232 1360 1568 864 865 868 911 +hsync -vsync

Option "dpms"
EndSection
保存然后重起试试看吧
十一.问：我的机器是windows和linux双系统，如何改变grub默认启动的系统？

答：这需要修改/boot/grub/grub.conf。举一个例子你就明白了。假设你的/boot/grub/grub.conf是这样子的：

default=0

timeout=10

splashimage=(hd0,7)/grub/splash.xpm.gz

title Red Hat Linux (2.4.18-14)

root (hd0,7)

kernel /vmlinuz-2.4.18-14 ro root=LABEL=/

initrd /initrd-2.4.18-14.img

title DOS

rootnoverify (hd0,0)

?1

那么你的grub会默认启动Red Hat Linux (2.4.18-14)这个系统，把default=0改成default=1，那么grub会
默认启动DOS这个系统。注意，这里的要点是：你想默认启动第n个title所指的系统，那

么default应该是
等于n-1
十二.问：我的文本控制台怎么总是出现乱码呢？
答：这是因为你安装了中文支持的缘故。解决的方法是安装一个zhcon(一个快速地外挂式CJK(中文/日文/韩文)
的多内码平台)，我把他放在附件中提供大家下载。关于zhcon的更进一步的消息，大家可以到他的官方主
页zhcon.gnuchina.org查看。安装和使用请参考这个
连接http://hepg.sdu.edu.cn/Service/tips/zhcon_manual.html
zhcon下载连接：http://zhcon.gnuchina.org/download/...on-0.2.1.tar.gz
十三.问：我在安装一个软件的时候，提示我缺少一个.so文件，安装无法继续，怎么办？
答：.so文件就像windows中的.dll文件一样，是库文件。一个程序的正常安装和运行需要特定的库文件
的支持。所以你需要去找到包含这个.so的包装上。去http://www.rpmfind.net
o文件?rpm包
十四.我访问windows分区时发现所有windows分区中的文件和文件夹名中的中文全变成问

号，怎么办？

答：在第三贴中我们讲解了通过编辑/etc/fstab实现在linux中访问windows的fat32分区

。同样，我们可
以通过进一步修改/etc/fstab来实现中文文件名显示。只要把/dev/hda1 /mnt/c vfat default 0 0中
的default全改为iocharset=cp936就行了。
十五.我的rh8.0中的XMMS不好使，不能播放MP3，怎么办？
答：这是因为rh公司怕别人告他侵权，所以在rh8.0中去掉了XMMS对MP3的支持，8.0以前

的版本都是好使的。
在8.0中要解决也很简单，装一个插件就行了。这个插件我放在本贴的附件里，rpm格式，经winrar压缩

附件：http://www.chinalinuxpub.com/vbbfor...s=&postid=86299
十六.问：我在linux中怎样才能使用windows分区呢？
答：先说一点背景知识 。linux支持很多种文件系统，包括windows的fat32和ntfs。对fat32的支持已经很好，
可以直接使用，而对ntfs的支持还不是太好，只能读，而写是极危险的，并且对ntfs的支持不是默认的，也就
是说你想要使用ntfs的话，需要重新编译内核。鉴于重编内核对于新手的复杂性，这里只讲解使用fat32分区的
方法 下面给出上述问题的两种解决方案：1.在安装系统（linux），进行到分区选择挂载点时，你可以建立几个
挂载点，如/mnt/c，/mnt/d等，然后选择你的windows fat32分区，把它们分别挂载到前

面建立的挂载点即可。
（注意，正如前面所说，在这里你不能把一个ntfs分区挂载到一个挂载点，应为ntfs不是默认支持的。）这样你
装好系统后就能直接使用你的windows fat32分区了。例如，你把windows的c盘（linux中的/dev/hda1）挂载
到/mnt/c，那么你就能在/mnt/c目录中找到你的c盘中的全部数据。2. 如果你在安装系统时没有像方案1所说
的那样挂载上你的fat32分区，没关系，仍然能够很方便的解决这个问题。首先，用一个

文本编辑器（如vi）打
开/etc/fstab，在文件的最后加入类似如下的几行
/dev/hda1 /mnt/c vfat default 0 0
你所要做的修改就是，把/dev/hda1改成你要挂载的fat32分区在linux中的设备号，把/mnt/c改成相应的挂载
点即可。注意，挂载点就是一个目录，这个目录要事先建立。举一个例子，我有三个fat32分区，在windows中
是c,d,e盘，在linux中的设备号分别为/dev/hda1,/dev/hda5,/dev/hda6。那么我就要先

建立3个挂载点，
如/mnt/c,/mnt/d,/mnt/e，然后在/etc/fstab中加上这么几行：
/dev/hda1 /mnt/c vfat default 0 0
/dev/hda5 /mnt/d vfat default 0 0
/dev/hda6 /mnt/e vfat default 0 0
?机器后就能直接使用c,d,e这三个fa
t32格式的windows分区了
十七.问：我的机器重装windows后，开机启动就直接进入了windows，原来的linux进不去了，怎么办？
答：这是由于windows的霸道。重装windows后，windows重写了你的mbr，覆盖掉了grub。解决方法
很简单：用你的linux第一张安装盘引导进入linx rescue模式（如何进入？你注意一下系统的提示信息
就知道了），执行下面两条命令就可以了
chroot /mnt/sysimage #改变你的根目录
grub-install /dev/hda #安装grub到mbr
十八.问：我的linux开机直接进入文本界面，怎样才能让它默认进入图形界面？

答：修改/etc/inittab文件，其中有一行id:3:initdefault，意思是说开机默认进入
运行级别3（多用户的文本界面），把它改成id:5:initdefault，既开机默认进入
运行级别5（多用户的图形界面）。这样就行了。
十九.如何同时启动多个x
以前的帖子,估计很多人没看过,贴出来温习一下
Linux里的X-Windows以其独特的面貌和强大的功能吸引了很多原先对linux不
感兴趣的人，特别是KDE和GNOME，
功能强大不说，而且自带了很多很棒的软件，界面非常友好，很适合于初学者。
下面告诉大家一个同时启
动6个X的小技巧：
在~/.bashrc中加入 以下几行：

```
alias X=startx -- -bpp 32 -quiet&
alias X1=startx -- :1 -bpp 32 -quiet&
alias X2=startx -- :2 -bpp 32 -quiet&
alias X3=startx -- :3 -bpp 32 -quiet&
alias X4=startx -- :4 -bpp 32 -quiet&
alias X5=startx -- :5 -bpp 32 -quiet&
```

其中32是显示器的色彩深度,你应该根据自己的实际情况设置。
之后运行

```
# bash
```

使改变生效，以后只要依次运行X,X1,X2,X3,X4,X5就可以启动6个X-Windows了。
二十.装了rpm的postgresql之后启动
/etc/init.d/postgresql start
是不能启动postgresql的tcp/ip连接支持的,所以打开/etc/init.d/postgresql这个文件

把

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl -D $PGDATA -p /usr/bin/postmas
ter start > /dev/null
2>&1" < /dev/null
```

改为:

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl -o -o -F -i -w -D $PGDATA -p /
usr/bin/postmaster start >
/dev/null2>&1" < /dev/null
```

这样就可以启动数据库的tcp/ip链接了
二十一.如何将man转存为文本文件
以ls的man为例

```
man ls |col -b >ls.txt
```

将info变成文本,以make为例

```
info make -o make.txt -s
```

二十二.如何在文本模式下发送2进制文件
首先检查系统有没有uuencode 和 uudecode如果没有从光盘上装

```
rpm -ivh sharutils-x.xx.x-x.rpm
```

假设要发送的文件是vpopmail-5.2.1.tar.gz执行

```
uuencode -m vpopmail-5.2.1.tar.gz vpopmail.tar.gz>encodefile
```

说明：
uuenode是编码命令，一m是使用mime64编码，vpopmail-5.2.1.tar.gz是要编码的文件，

vpopmail.tar.gz是如
果解码后得到的文件名，encodefile是编码后的文件名。
执行上述命令之后就可以通过mail命令发送编码后的文件了
mail chenlf@chinalinuxpub.comN 1 chenlf@ns1.catv.net Mon Jun 10 16:4??牴????
?4 17/363
N 2 root@ns2.catv.net Mon Jun 10 16:45 6091/371145
& 2
Message 2:
From root@ns2.catv.net Mon Jun 10 16:45:28 2002
Date: Mon, 10 Jun 2002 16:44:51 +0800
From: root
To: chenlf@chinalinuxpub.com
begin-base64 644 vpopmai.tar.gz
H4sIABr15TwAA+w9a2PbNpL7NfwVqNPbWIlFPSzbiR2n9SuxE7/OcuLNtdmU
EiGLMUWqfFhWt7u//eYBgKRE2U7iTa+3VndjiQQGg5nBYDAYDC6H4XDgeH51
yW7ajdpf/h2fer1VX1lagr/1+spyq/BXff5SX2mtNBZXmovN5l/qjWZrqfEX
sfRvwWbik8aJEwnxl7ifDofXlLvp/Z/0c1nk/8uN/777NuqNen25lZrB/+XF
pcUG8r/ZbC0vL9ZXoPwi/08von73qEx//sP5bwHHxanT8aUIe2IrDBIZJLFl
7QVJFFovpZ0kkYxFL4yEFhVLCKhk1W2xG45E1wnEnohlIsJAiksvSlLHF24I
JQ0RhKIjRdKXYhh5Ayca6xcAD8DQm4HT7XuB/EGcSXgbPErEyAkSrNp3LqVw
grGoyaRbGzpxPHJFGssotq0Gtw619gTgJbixode9E0lQDMaTmEjE/AerydVc
rAY4jJzIFY7vC3wL2DgJvJIxIjFwkm6fWkfw1KoAIti/EgkWc3A6YRp05ReB
aeXAQH34GoXOwAvOVUnoEnwRYRqJeJAMgczRpYzEyEv6YQoUH8oAClLtjjD
Rr1YOCJ2BkPgJop1IuJu5A0TYh9xIdQwfrCWTdt9pMKvaZg4j5jT3PgojC5+
sFZswMOLAJzvSyhGXQSCOmLo09DtEOAicBCD2qUT1agAg44BSd+1niIEzVPs
....
....
....
& s 2 encodefile
"encode" [New file]
& q
然后进行解码
uudecode encodefile
ls
encodefile vpopmai.tar.gz
tar zxvf vpopmail.tar.gz
OK了
二十三.将 man page 转成 HTML 格式
使用 man2html 这个指令，就可以将 man page 转成 HTML 格式了。用法是:
man2html filename > htmlfile.html
二十四.如何在gnome和kde之间切换。
如果你是以图形登录方式登录linux，那么点击登录界面上的session（任务）即可以选择gnome和kde。
如果你是以文本方式登录，那执行switchdesk gnome或switchdesk kde，然后再startx就可以进入gnome或kde。
25...tar，.tar.gz，.bz2，.tar.bz2，.bz，.gz是什么文件，如何解开他们?
他们都是文件(压缩)包。

.tar：把文件打包，不压缩：tar cvf *.tar dirName 解开：tar xvf *.tar
.tar.gz：把文件打包并压缩：tar czvf *.tar.gz dirName 解开：tar xzvf *.tar.gz

.bz2：解开：bzip2 -d *.bz2
.bz：解开：bzip -d *.bz
.gz：解开：gzip -d *.gz
26.linux下如何解开.zip，.rar压缩文件？
rh8下有一个图形界面的软件file-roller可以做这件事。令外可以用unzip *.zip解开zip文件，unrar *.rar解
开rar文件，不过unrar一般系统不自带，要到网上下载。
27.linux下如何浏览.iso光盘镜像文件？
a.建一个目录，如：mkdir a
b.把iso文件挂载到该目录上：mount -o loop xxxx.iso a
现在目录a里的内容就是iso文件里的内容了。
28.linux下如何配置网络？
用netconfig。“IP address:”就是要配置的IP地址，子网掩
码，“Default gateway (IP):”网关，“Primary nameserver:”DNS服务器IP。
29.如何让鼠标支持滚轮？
在配置鼠标时，选择微软的鼠标，并正确选择端口如ps2,usb等
30.如何让控制台支持中文显示？
安装zhcon。zhcon需要libimm_server.so和libpth.so.13这两个库支持。
一般的中文输入法应该都有libimm_server.so。libpth.so.13出自pth-1.3.x。把这两个

文件放到/usr/lib下就行了。
31.如何配置grub？
修改/boot/grub/grub.conf文件。其中
“default=n”(n是个数字)是grub引导菜单默认被选中的项，n从0开始，0表示第一项，

1表示第二项，依此类推。
“timeout=x”(x是一个数)是超时时间，单位是妙。也就是引导菜单显示后，如果x秒内

用户不进行选择，
那么grub将启动默认项。
“splashimage =xxxxxx”，这是引导菜单的背景图，先不理他。
其它常用项我用下面的例子来说明：
title Red Hat 8.0
root (hd1,6)
kernel /boot/vmlinuz-2.4.18-14 ro root=/dev/hdb7
initrd /boot/initrd-2.4.18-14.img
其中"Red Hat 8.0"是在启动菜单列表里显示的名字
root (hdx,y)用来指定你的boot分区位置，如果你没有分boot分区（本例就没分boot分区），那就指向根分区
就行了，hdx是linux所在硬盘，hd0是第一块硬盘，hd1是第二块，依此类推。y是分区位

置，从0开始，也就是
等于分区号减一，比如你要指向的分区是hdx7，那么y就是6，如果是hdx1，那y就是0。
注意root后面要有一个

空格。
kernel /boot/vmlinuz-2.4.18-14，其中"/boot/vmlinuz-2.4.18-14"是你要用的内核路

径，如果你编译了心内
核，把它改成你的新内核的路径就行了。
ro就不用管，写上不会有错。
root=/dev/hdxx指定根分区，本例是hdb7，所以root=/dev/hdb7
initrd xxxxxxxxxxxxx这行不要也行，目前我还不清楚它是做什么用的。
上面是linux的，下面是windows的
title windows 98
rootnoverify (hd0,0)
chainloader +1
title xxxxxxx不用解释了，上面有解释。
rootnoverify (hdx,y)用来指定windows所在分区，x，y跟上面一样，注意rootnoverif
y后有空格。
chainloader +1照抄就行，注意空格。
32.如何配置lilo?
修改/etc/lilo.conf文件，然后运行lilo命令。
33.如何让linux启动后自动挂载制定分区?
编辑/etc/fstab文件，在里面加入相应的行，如果你对/etc/fstab的配置不熟悉，你可
以用ptm，一个自动配
置fstab的软件，在公社里应该有下载。
34.如何定制linux启动时启动的服务?
在控制台或终端下运行ntsysv，然后把要启动的服务选上，如果要开机不启动某个那就
把他的选中去掉。
35.没有软驱如何修复grub/lilo引导菜单?
a.把第一张linux安装盘里的dosutils目录复制到windows盘中。如果是iso可以用winra
r3提取。
b.进入纯dos，进入dosutils目录，执行loadlin autoboot/vmlinuz root=/dev/hdxx()
hdxx是你的linux根分区。
这样就能进入linux。
c.执行grub-install /dev/hdx(x=a,b,c,d) 或lilo即可以重写引导。
另外还可以参考：http://www.linuxfans.org/nuke/modul...iewtopic&t=5101
36.如何配置鼠标?
打开一个终端或进入控制台，运行mouseconfig命令即可配置鼠标!
37.如何从dos下进入linux?
首先你需要loadlin.exe这个程序，linux第一张安装盘里的dosutils目录里有，另外你
需要的就是linux内
核，/boot/vmlinuz-2.4.xx-xx就是，把他弄到windows盘中，并把他重命名为vmlinuz（

主
??牴??? ?????
要是vmlinuz-2.4.xx-xx在dos下显示有点问题），另外dosutils目录里的autoboot里也
有一
个vmlinuz，也是linux内核，凑合能用，但很多设备不支持，如果linux引导程序或lin
ux启动配置
有问题，进不了linux，可以用这个内核进去修复linux。

然后用loadlin vmlinuz root=/dev/hdxx （hdxx是你的linux根分区），另外
用loadlin vmlinuz root=/dev/hdxx single还可以实现单用户模式。
38.root密码丢失解决
GRUB:在引导装载程序菜单上，键入 [e] 来进入编辑模式。
你会面对一个引导项目列表。查找其中类似以下输出 的句行：
kernel /vmlinuz-2.4.18-0.4 ro root=/dev/hda2
按箭头键直到这一行被突出显示，然后按 [e]。
你现在可在文本结尾处空一格再添加 single 来告诉 GRUB 引导单用户 Linux 模式。按

[Enter] 键来
使编辑结果生效。
你会被带会编辑模式屏幕，从这里，按 ， GRUB 就会引导单用户 Linux 模式。 结束载

入后，你会面对
一个与以下相似的 shell 提示：
sh-2.05#
现在，你便可以改变根命令，键入：
bash# passwd root
你会被要求重新键入口令来校验。结束后， 口令就会被改变，你便可以在提示下键入
reboot 来重新引导；
然后，象平常一样登录为根用户。
LILO:当系统启动到出现LILO引导画面时，对于图形引导方式按TAB键进入文本方式，然
后在LILO处
输入linux single回车即可进入免密码的控制台，进入以后使用passwd命令修改root的
密码
即可
0001
修改主机名
vi /etc/sysconfig/network，修改HOSTNAME一行为HOSTNAME=主机名，重启后也能生效

0002
Ret Hat Linux启动到文字界面(不启动xwindow)
将/etc/inittab中 id:5:initdefault: 一行中的5改为3
0003
redhat的自动升级更新问题(hutuworm)
在http://www.redhat.com/corp/support/errata/...
升级除kernel外的rpm: up2date -u
升级包括kernel在内的rpm: up2date -u -f
0004
windows下看linux分区的软件
Paragon.Ext2FS.Anywhere.2.5.rar和explore2fs-1.00-pre4.zip
0005
mount用法
fat32的分区 mount -o codepage=936,iocharset=cp936 /dev/hda7 /mnt/cdrom
iso文件 mount -o loop /abc.iso /mnt/cdrom
软盘 mount /dev/fd0 /mnt/floppy
USB闪存 mount /dev/sda1 /mnt/cdrom

所有/etc/fstab内容 mount -a
可以指定文件格式"-t 格式", 格式可以为vfat, ext2, ext3等.
0006
在vmware的LINUX中使用本地硬盘的FAT分区
将本地的FAT分区共享，然后在VMWARE中使用SMBFS挂上。可以将如下的行放到/etc/fstab中：
_ip/D$ /mnt/d smbfs defaults,auto,username=win_name,password=win_pass,codepage=936,
iocharest=gb2312 0 0
其中win_ip是你的windows的IP地址；
D$是你的windows里面共享的D盘的共享名；
/mnt/d是要将该分区mount到linux的目录；
win_name和win_pass是你的WINDOWS里面可以读取该分区的用户，比如你的管理员名和密

码。
如果你运行了/etc/rc.d/init.d/netfs，那么在启动的时候就会自动挂载这个分区。
0007.a
删除名为-a的文件
rm ./-a
rm -- -a 告诉rm这是最后一个选项，参见getopt
ls -i 列出inum，然后用find . -inum inum_of_thisfile -exec rm {} \;
0007.b
?文件
rm \\a
0007.c
删除名字带的/和‘\0文件
这些字符是正常文件系统所不允许的字符，但可能在文件名中产生，如unix下的NFS文件

系统在Mac系统上使用
1.解决的方法，把NFS文件系统在挂到不过滤/字符的系统下删除含特殊文件名的文件。

2.也可，将错误文件名的目录其它文件移走，ls -id 显示含该文件目录的inum，umount 文件系统，
clri清除该目录的inum，fsck，mount，check your lost+found，rename the file in it.
最好是通过WINDOWS FTP过去就可以删除任何文件名的文件了！
0007.d
删除名字带不可见字符的文件
列出文件名并转储到文件：ls -l >aaa
然后编辑文件的内容加入rm命令使其内容成为删除上述文件的格式：
vi aaa
[rm -r *******
]
把文件加上执行权限 chmod +x aaa
执行 $aaa
0007.e
删除文件大小为零的文件

rm -i `find ./ -size 0`
find ./ -size 0 -exec rm {} \;
find ./ -size |xargs rm -f &非常有效
for file in * #自己定义需要删除的文件类型
do
if [! -s ${file}]
then
rm ${file}
echo "rm $file Success!"
fi
done
0008
redhat设置滚轮鼠标(mc1011)
进入X后，选择鼠标的配置，选择wheel mouse(ps/2)就可以耍?
如果鼠标表现异常，重启计算机即可。
0009
加装xwindow
用linux光盘启动，选择升级，然后单独选择包，安装即可
0010
删除linux分区
做一张partition magic的启动软盘,启动后删除. 或者用win2000的启动光盘启动,然后删除.
0011
如何退出man
q
0012
不编译内核，mount ntfs分区
原装rh8，未升级或编译内核
1. 上google.com搜索并下载 kernel-ntfs-2.4.18-14.i686.rpm
2. rpm -ivh kernel-ntfs-2.4.18-14.i686.rpm
3. mkdir /mnt/c
4. mount -t ntfs /dev/hda1 /mnt/c
0013
redhat 8.0中用xmms听mp3
下载http://www.gurulabs.com/files/xmms-mp3-1.2.7-13.p.i386.rpm
rpm -Uvh xmms-mp3-1.2.7-13.p.i386.rpm
0014
找回忘记了的root口令（lilo/grub）
三种办法：
1.在系统进入单用户状态，直接用passwd root去更改
2.用安装光盘引导系统，进行linux rescue状态，将原来/分区挂接上来,作法如下：
cd /mnt
mkdir hd
mount -t auto /dev/hdaX(原来/分区所在的分区号） hd
cd hd
chroot ./
passwd root

这样可以搞定
3.将本机的硬盘拿下来，挂到其他的linux系统上，采用的办法与第二种相同
rh8中
一. lilo
1. 在出现 lilo: 提示时键入 linux single
画面显示 lilo: linux single
2. 回车可直接进入linux命令行
3. #vi /etc/shadow
将第一行，即以root开头的一行中root:后和下一个:前的内容删除，
第一行将类似于
root::......
保存
4. #reboot重启，root密码为空
二. grub
1. 在出现grub画面时，用上下键选中你平时启动linux的那一项(别选dos哟)，然后按e键
2. 再次用上下键选中你平时启动linux的那一项(类似于kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/)，
然后按e键
3. 修改你现在见到的命令行，加入single，结果如下：
kernel /boot/vmlinuz-2.4.18-14 single ro root=LABEL=/
4. 回车返回，然后按b键启动，即可直接进入linux命令行
?i /etc/shadow
将第一行，即以root开头的一行中root:后和下一个:前的内容删除，
第一行将类似于
root::......
保存
6. #reboot重启，root密码为空
0015
使ctrl + alt + del失效
vi /etc/inittab
将ca::ctrlaltdel:/sbin/shutdown -t3 -r now这行注释掉，就可以了
0016
如何看出redhat的版本是7还是8(hutuworm)
cat /proc/version或者cat /etc/redhat-release或者cat /etc/issue
0017
文件在哪个rpm中(无双)
上http://www.rpmfind.net-qf 文件名得到
0018
把man或info的信息存为文本文件
以 tcsh 为例：
man tcsh | col -b > tcsh.txt

Linux 挂载文件系统

1、挂载点必须是一个目录。

2、一个分区挂载在一个已存在的目录上，这个目录可以不为空，但挂载后这个目录下以前的内容将不可用。
对于其他操作系统建立的文件系统的挂载也是这样。但是需要理解的是：光盘、软盘、其他操作系统使用的文件系统的格式与linux使用的文件系统格式是不一样的。光盘是ISO9660；软盘是fat16或ext2；windows NT是fat16、NTFS；windows98是fat16、fat32；windows2000和windowsXP是fat16、fat32、NTFS。挂载前要了解linux是否支持所要挂载的文件系统格式。

挂载时使用mount命令：
格式：mount [-参数] [设备名称] [挂载点]

其中常用的参数有
-t<文件系统类型> 指定设备的文件系统类型，常见的有：
minix linux最早使用的文件系统
ext2 linux目前常用的文件系统
msdos MS-DOS的fat，就是fat16
vfat windows98常用的fat32
nfs 网络文件系统
iso9660 CD-ROM光盘标准文件系统
ntfs windows NT 2000的文件系统
hpfs OS/2文件系统
auto 自动检测文件系统
-o<选项> 指定挂载文件系统时的选项。有些也可用在/etc/fstab中。常用的有
codepage=XXX 代码页
iocharset=XXX 字符集
ro 以只读方式挂载
rw 以读写方式挂载
nouser 使一般用户无法挂载
user 可以让一般用户挂载设备

提醒一下，mount命令没有建立挂载点的功能，因此你应该确保执行mount命令时，挂载点已经存在。（不懂？说白了点就是你要把文件系统挂载到哪，首先要先建上个目录。这样OK？）

例子：windows98装在hda1分区，同时计算机上还有软盘和光盘需要挂载。

```
# mk /mnt/winc
# mk /mnt/floppy
# mk /mnt/cdrom
# mount -t vfat /dev/hda1 /mnt/winc
# mount -t msdos /dev/fd0 /mnt/floppy
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

现在就可以进入/mnt/winc等目录读写这些文件系统了。

要保证最后两行的命令不出错，要确保软驱和光驱里有盘。（要是硬盘的磁盘片也可以经常随时更换的话，我想就不会犯这样的错误了 :-> ）
如果你的windows98目录里有中文文件名，使用上面的命令挂载后，显示的是一堆乱码。这就要用到 -o 参数里的codepage iocharset选项。codepage指定文件系统的代码页，简体中文中文代码是936；iocharset指定字符集，简体中文一般用cp936或gb2312。

当挂载的文件系统linux不支持时，mount一定报错，如windows2000的ntfs文件系统。可以重新编译linux内核以获得对该文件系统的支持。关于重新编译linux内核，就不在这里说了。

四 、自动挂载
每次开机访问windows分区都要运行mount命令显然太烦琐，为什么访问其他的linux分区不用使用mount命令呢？
其实，每次开机时，linux自动将需要挂载的linux分区挂载上了。那么我们是不是可以设定让linux在启动的时候也挂载我们希望挂载的分区，如windows分区，以实现文件系统的自动挂载呢？
这是完全可以的。在/etc目录下有个fstab文件，它里面列出了linux开机时自动挂载的文件系统的列表。我的/etc/fstab文件如下：

```
/dev/hda2 / ext3 defaults 1 1
/dev/hda1 /boot ext3 defaults 1 2
none /dev/pts devpts gid=5,mode=620 0 0
none /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0
/dev/hda3 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,codepage=936,iocharset=gb2312 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
/dev/hdb1 /mnt/winc vfat defaults,codepage=936,iocharset=cp936 0 0
/dev/hda5 /mnt/wind vfat defaults,codepage=936,iocharset=cp936 0 0
```

在/etc/fstab文件里，第一列是挂载的文件系统的设备名，第二列是挂载点，第三列是挂载的文件系统类型，第四列是挂载的选项，选项间用逗号分隔。第五六列不知道是什么意思，还望高手指点。

在最后两行是我手工添加的windows下的C；D盘，加了codepage=936和iocharset=cp936参数以支持中文文件名。参数defaults实际上包含了一组默认参数：

rw 以可读写模式挂载
suid 开启用户ID和群组ID设置位
dev 可解读文件系统上的字符或区块设备
exec 可执行二进制文件
auto 自动挂载
nouser 使一般用户无法挂载
async 以非同步方式执行文件系统的输入输出操作

大家可以看到在这个列表里，光驱和软驱是不自动挂载的，参数设置为noauto。（如果你非要设成自动挂载，你要确保每次开机时你的光驱和软驱里都要有盘，呵呵。）
<hr>

上一页 目录 下一页

Linux 使用技巧集锦(二)

1. 重装Win9x后lilo失效无法启动Linux

只须先用软盘启动Linux然后运行一次lilo即可。对于RedHat可以用命令mkbootdisk来制作启动盘，例如：

mkbootdisk --devices /dev/fd0 2.0.34-1

其中2.0.34-1是 /lib/modules下的一个目录。

2. 把dos/windows改为缺省启动的OS

RedHat把Linux作为lilo缺省启动的OS，即出现 boot: 提示符后直接回车启动Linux。如果想改为dos/Windows，修改文件/etc/lilo.conf，把有关启动dos的部分放在前面

，再运行一次lilo即可。例如把lilo.conf从：

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.34-1
label=linux
root=/dev/hdc5
read-only
other=/dev/hda1
label=dos
table=/dev/hda
```

改为：

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
other=/dev/hda1
label=dos
table=/dev/hda
image=/boot/vmlinuz-2.0.34-1
label=linux
root=/dev/hdc5
read-only
```

3. 去掉引导区内的lilo信息

安装lilo后如果硬盘数量或分区情况有改变的话将导致硬盘不能启动，这时只须用软盘启动dos再运行 fdisk/mbr 即可。(这个方法还可清除任何引导区病毒)

4. 让 X 支持 AGP 显卡

下载一个 AX 4.1 即 AcceleratedX 4.1 ， 一个商业版的 X Server 。除 AGP 显卡外 AX 还支持很多牌号的显卡。另外还可以用AX 带的设置程序 Xsetup 方便的对分辨

率，刷新率等参数进行设置。不过我也不知道用AGP卡效果如何，因为我现在还在用老掉

牙的 S3 765 。

5. 把 man 或 info 的信息存为文本文件

以 tcsh 为例： man tcsh | col -b > tcsh.txt

info tcsh -o tcsh.txt -s

6. 用当前路径作提示符

对 bash 来说，在 .bashrc 里加一行：

PS1="\$PWD\\$"

对 tcsh 来说，在 .tcshrc 里加一行：

set prompt="%/>"

7. 压缩可执行文件

Linux 下有一个类似 dos 里的 pklite 和 lzexe 的命令 -- gzexe 。而且压缩率一般都可以超过 50% ，在空间紧张时很有用。比如：

```
/dosc/temp# ls -al
total 148
-rwxr-xr-x 1 root root 149564 Dec 8 15:33 gawk
/dosc/temp# gzexe gawk
gawk: 54.5%
/dosc/temp# ls -al
total 216
-rwxr-xr-x 1 root root 68710 Dec 8 15:36 gawk
-rwxr-xr-x 1 root root 149564 Dec 8 15:33 gawk~
/dosc/temp#
```

8. 查看 Linux 启动时的信息

Linux 启动时屏幕显示的信息来不及看清就一闪而过，如果对这些信息感兴趣的话可以在启动完后用命令 dmesg 查看。

9. 处理文件名内含有特殊字符的文件

如果有一个文件名叫 -file 如果想删除它，键入 rm -file 会显示 invalid option ，原来由于文件名的第一个字符为 - ， Linux 把文件名当作选项了，可以加 -- 解

决这个问题， 如 rm -- -file 。如果是其他特殊字符的话可以在特殊字符前加一个 \ ，或者用双引号把整个文件名括起来。

10. 一次处理一整个目录

Linux/UNIX 的很多常用命令如 rm ， cp 等都有一个参数---- -r ， 是递归的意思， 命令里加了参数 -r 就可以对目标目录及其下所有子目录进行操作，如：

rm -rf /test (f 是 force 意为强行) ， 该命令完全删除根目录下的子目录 test ，作用类似于 dos 下的 deltree ，当然使用这个命令时要特别小心。再如：

cp -r /test /test1 有类似 dos 下 xcopy /s 的作用。

11. 修改登录画面

对本机：
先修改文件 /etc/issue 改为相要显示的内容， 再修改文件 /etc/rc.d/rc.local(Red Hat) 或 /etc/rc.d/rc.S(Slackware) 把下面几句注释掉：

```
# This will overwrite /etc/issue at every boot. So, make any changes you
# want to make to /etc/issue here or you will lose them when you reboot.
echo "" > /etc/issue
echo "Red Hat Linux $R" >> /etc/issue
echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
cp -f /etc/issue /etc/issue.net
echo >> /etc/issue
```

不然的话每次重新启动 /etc/issue 都会被更改。
对 telnet 的远地机器：
先把文件 /etc/usr/sbin/in.telnetd 改名，如改为 in.telnetd.exe ，再编一个名为 in.telnetd 的 shell 脚本， 在显示完需要的内容后再调用 in.telnetd.exe 。如：

```
#!/bin/sh
cat /etc/login.banner #需要在登录提示符前显示的内容
echo -n ""
exec /usr/sbin/in.telnetd.exe
```

12. 几个有用的别名
几个能方便操作和减少误操作的别名，建议把它们放到启动文件里， 如 /etc/bashrc 。

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
#前三个别名的作用是在删除、覆盖文件之前先提示确认，RedHat已经自动
#加上了，强烈建议Slackware的用户也加上，因为Linux/UNIX下文件一旦删
#除就再也无法恢复了。
alias l='ls -l'
alias cd..='cd ..'
alias utar='tar xvfz' #解 *.tar.gz 文件
alias inforpm='rpm -qpi' #显示rpm包的信息
alias instrpm='rpm -Uhv' #安装rpm包
alias listrpm='rpm -qpl' #列rpm包内的文件
alias uistrpm='rpm -e' #反安装rpm包
```

13. 使用命令补齐
所谓命令补齐(Command-Line Completion)是指当键入的字符足以确定目录下一个唯一的文件时只须按 Tab 键就可以自动补齐该文件名的剩下部分，例如要把目录 /freesoft 下的文件 gcc-2.8.1.tar.gz 解包，当键入到 tar xvfz /freesoft/g 时如果该文件是该目录下唯一以 g 起头的文件的话就可以按下 Tab 键，这时命令会被自动补齐为：tar xvfz /freesoft/gcc-2.8.1.tar.gz ，非常方便。
14. 最后一条技巧
有时间的话多看看系统提供的FAQ文件，因为通常遇到的问题大多数都能在里面找到答案。

上一页 目录 下一页

Linux 使用技巧集锦(一)

初用Linux时可能有处处不方便的感觉, 可是等使用一段时间掌握了一些技巧后就会

感到越来越顺手了. 以下就是我使用Linux以来积累的一些经验与技巧, 相信对初学者会

有所帮助的.

1. 使用虚拟控制台

登录后按Alt+F2键这时又可以看到"login:"提示符, 这个就是第二个虚拟控制台. 一般新安装的Linux有四个虚拟控制台, 可以用Alt+F1~Alt+F4来访问. 虚拟控制台最有用的时候是当一个程序出错锁住输入时可以切换到其他虚拟控制台登录进入后杀掉这个出错的进程.

2. 拷贝与粘贴

字符界面: 不管是Slackware 还是RedHat 安装后每次启动时都会自动运行一个叫gpm的程序, 该程序运行后就可以用鼠标来拷贝与粘贴了. 具体做法是按住鼠标左键拖动使要拷贝的地方反白, 这时反白的区域已经被拷贝, 再按鼠标右键拷贝的内容就会被粘贴在光标所在位置了.

X-Window下: 拷贝的操作与字符界面下一样, 三键鼠标的话按中键粘贴, 两键鼠标的话同时按左右键粘贴(须在配置XF86时在鼠标的设置里选择 Emulate 3 Button).

3. 快速进入某些目录

键入 cd ~ 可进入用户的home目录.

键入 cd - 可进入上一个进入的目录.

4. 软驱的使用

如果是Linux 的ext2文件系统用如下命令:

#mount -t ext2 /dev/fd0 /mnt

dos格式的软盘则用命令:

#mount -t msdos /dev/fd0 /mnt

然后就可以在/mnt里访问软盘的内容了, 注意在取出软盘之前要先卸掉软盘上的文件系统,

#umount /mnt 否则会导致信息丢失.

在软盘上建立文件系统可用如下命令:

#mke2fs /dev/fd0 1440

5. RedHat下显示彩色目录列表

打开/etc/bashrc, 加入如下一行:

alias ls="ls --color"

下次启动bash时就可以像在Slackware里那样显示彩色的目录列表了, 其中颜色的含义如

下:

蓝色-->目录
绿色-->可执行文件
红色-->压缩文件
浅蓝色-->链接文件
灰色-->其他文件
6. 显示文件的类型
用命令 file 可以使你知道某个文件究竟是ELF格式的可执行文件，还是shell script文件或是其他的什么格式 例如:
#file startx
7 .命令文件的路径
当你键入一些命令如find, shutdown, mount, startx时是否想过要知道这些文件究竟放

在哪个目录里? 用命令
whereis 可以实现,如
#whereis startx
8. 查找文件
find 命令要加一些参数才能查到文件, 如:
#find /usr -name XF86Setup -print
在/usr里查找文件XF86Setup.
9. 删除无用的core文件
程序运行出后有时候会产生一个名为core的文件, 这个文件一般很大而且没什么用, 可以删除掉以释放空间.
10. 一次解开.tar.gz文件
解开.tar.gz文件一般要先用gunzip解压再用tar解包, 在Linux下可以一次完成, 如:
#tar zxpvf sample.tar.gz
11. 显示win95分区里的长文件名
如果你发现win95分区里的长文件命不能显示, 可以重新用vfat方式来mount. 对于启动时就mount的分区可以修改文件/etc/fstab, 将里面的msdos字样改为vfat. 如果无法用vfat mount, 则要重新编译一下核心, 加入对vfat的支持.
12. Linux 里的Norton Commander
在提示符下键入命令mc后你就会看到一个与Norton Commander很相似的界面, 实际上功能也很相似甚至更强大, 比如可以直接对.tar.gz 压缩包里的文件进行操作(有点像

ZipMagic).
13. 启动后直接进入X
编辑/etc/inittab文件, 把id:3:initdefautl改为 id:4:initdefautl. 不过搜索路

径可能会有些问题, 关机时你得直接进入/sbin 里去执行 shutdown.
14. 后台运行 X 程序
执行一个X程序时别忘了在后面加一个&号,如 #netscape & 否则的话在你推出执行的程序之前再无法在那个终端窗口里输入命令了.
15. 强行退出 X
有时候在 X 里由于程序出错鼠标键盘都不起作用, 这时候不用着急, 因为在Linux下几乎不会像在Win95里那样恶性死机, 你只须键入Ctrl+Alt+BackSpace键就可以回到字

符界面下了.

上一页 目录 下一页

LINUX 下访问别的分区

用fdisk -l列出本硬盘的所有分区，下面是我的：

[root@magic root]# fdisk -l

Disk /dev/hdb: 255 heads, 63 sectors, 4866 cylinders

Units = cylinders of 16065 * 512 bytes

Device Boot Start End Blocks Id System

/dev/hdb1 * 1 637 5116671 7 HPFS/NTFS

/dev/hdb2 638 4865 33961410 f Win95 Ext'd (LBA)

/dev/hdb5 638 1657 8193118+ 7 HPFS/NTFS

/dev/hdb6 1658 2294 5116671 7 HPFS/NTFS

/dev/hdb7 2295 2931 5116671 b Win95 FAT32

/dev/hdb8 2932 3568 5116671 b Win95 FAT32

/dev/hdb9 3569 4218 5221093+ b Win95 FAT32

/dev/hdb10 4219 4231 104391 83 Linux

/dev/hdb11 4232 4800 4570461 83 Linux

/dev/hdb12 4801 4865 522081 82 Linux swap

其中：/dev/hdb5是WINDOWS的D盘，依此类推

/dev/hdb6——E盘

/dev/hdb7——F盘

/dev/hdb8——G盘

/dev/hdb9——F盘

具体如何挂装这WINDOW分区，可以看如下文章(这是在本版精华区)：

http://www.linuxfans.org/nuke/modules.php?name=Forums&file=viewtopic&t=5907

当然用这个也没关系:

mount -o rw -t vfat /dev/hdb7 /mnt/f codepage=936 iocharset=cp936

这样就可以把/dev/hdb7这个在WINDOWS下为F盘挂装到了LINUX下的/mnt/f里,不过,/mnt/f这个目录要事先用mkdir /mnt/f建好

下面是我的/etc/fstab的内容:

LABEL=/ / ext3 defaults 1 1

LABEL=/boot /boot ext3 defaults 1 2

none /dev/pts devpts gid=5,mode=620 0 0

none /proc proc defaults 0 0

none /dev/shm tmpfs defaults 0 0

/dev/hdb12 swap swap defaults 0 0

/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0

/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
/dev/hdb7 /mnt/f vfat rw,codepage=936,iocharset=cp936 0 0
/dev/hdb8 /mnt/g vfat rw,codepage=936,iocharset=cp936 0 0
/dev/hdb9 /mnt/h vfat rw,codepage=936,iocharset=cp936 0 0
其中加粗部分是手工添加的,这样在系统启动的时候自动可以挂装。当然把mount -o rw -t vfat /dev/hdb7 /mnt/f codepage=936 iocharset=cp936这句话添加到/etc/rc.local的最后面，然后保存，重起，就可以自动访问WINDOWS的F盘了。
注，加上rw参数，普通用户也可以访问。上面的操作一定要在root用户状态下操作。

iocharset是设置支持的语言,cp936应该是支持中文,不过我是写iocharset=gb2312,应该一样的.而后面的两个数值分别代表Dump frequency和fsck检查顺序.Dump frequency是指能否用dump工具进行备份以及频率是多少(0表示不能).fsck检查顺序表示开机运行时是否用fsck进行检查以及顺序的次序.一般windows的分区都是0 0,而只有在linux分区时,才要用到这两个选项.
而/etc/fstab这个文件就是系统挂载的依据.还有两个重要选项,user 和auto.user代表用户可以挂载,auto是指可以用mount -a自动挂载(系统开机时会执行这个命令).下面是我系统的/etc/fstab文件:
LABEL=/ / ext3 defaults 1 1
none /dev/pts devpts gid=5,mode=620 0 0
none /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0
/dev/hda9 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
/dev/hda1 /mnt/c auto auto,rw,iocharset=gb2312 0 0
/dev/hda5 /mnt/d auto auto,rw,iocharset=gb2312 0 0
/dev/hda6 /mnt/e auto auto,rw,iocharset=gb2312 0 0
/dev/hda7 /mnt/f auto auto,rw,iocharset=gb2312 0 0
/dev/hda10 /ftp auto defaults 1 1
<hr>

上一页 目录 下一页

上一页 目录 下一页

Linux常用命令简介

su
su命令是最基本的命令之一，常用于不同用户间切换。例如，如果登录为 user1，要切换为user2，只要用如下命令：
$su user2
然后系统提示输入user2口令，输入正确的口令之后就可以切换到user2。完成之后就可以用exit命令返回到user1。
su命令的常见用法是变成根用户或超级用户。如果发出不带用户名的su命令 ，则系统提示输入根口令，输入之后则可切换为根用户。

如果登录为根用户，则可以用su命令成为系统上任何用户而不需要口令。

pwd
pwd命令也是最常用最基本的命令之一，用于显示用户当前所在的目录。

cd
cd命令不仅显示当前状态，还改变当前状态，它的用发跟dos下的cd命令基本一致。
cd ..可进入上一层目录
cd -可进入上一个进入的目录
cd ~可进入用户的home目录

ls
ls命令跟dos下的dir命令一样，用于显示当前目录的内容。
如果想取得详细的信息，可用ls -l命令， 这样就可以显示目录内容的详细信息。
如果目录下的文件太多，用一屏显示不了，可以用ls -l |more分屏显示 。

find
find命令用于查找文件。这个命令可以按文件名、建立或修改日期、所有者(通常是建立文件的用户)、文件长度或文件类型进行搜索。
find命令的基本结构如下：
$find
其中指定从哪个目录开始搜索。指定搜索条件。表示找到文件怎么处理。一般来说，要用-print动作，显示 整个文件路径和名称。如果没有这个动作，则find命令进行所要搜索而不显示结果，等于白费劲。

例如，要搜索系统上所有名称为ye的文件，可用如下命令：
$find / -name ye -print
这样就可以显示出系统上所有名称为ye的文件。

tar
tar最初用于建立磁带备份系统，目前广泛用于建立文件发布档案。可用如下方法建立tar档案：
$tar cvf
例如，如果要将当前目录中所有文件存档到ye.tar中，可用如下命令：
$tar cvf ye.tar *.*
要浏览档案内容，将c选项变成t。如果要浏览ye.tar档案中的内容，可用如下命令：
$tar tvf ye.tar
要取出档案内的内容，将c选项变成x。如果要将ye.tar档案中的内容取到当前目录中，可用如下命令：
$tar xvf ye.tar

gzip
gzip命令用于压缩文件。 例如，如果要将ye.txt文件压缩，可用如下命令：
$gzip ye.txt
这样就可以压缩文件并在文件名后面加上gz扩展名，变成文件ye.txt.gz。
解压缩文件可用gzip -d命令实现：
$gzip -d ye.txt.gz
这样就可以解压缩文件并删除gz扩展名。除此之外还可以用gunzip命令来解 压缩文件，效果跟用gzip -d命令一样。
旧版的tar命令不压缩档案，可用gzip压缩。例如：

$tar cvf ye.tar *.txt
$gzip ye.tar
则可建立压缩档案ye.tar.gz。
新版的tar可以直接访问和建立gzip压缩的tar档案，只要在tar命令中加上z 选项就可以了。例如：
$tar czvf ye.tar *.txt
生成压缩档案ye.tar.gz，
$tar tzvf ye.tar *.txt
显示压缩档案ye.tar.gz的内容，而
$tar xzvf ye.tar *.txt
取出压缩档案ye.tar.gz的内容。

mkdir
这个命令很简单，跟dos的md命令用法几乎一样，用于建立目录。

cp
cp命令用于复制文件或目录。
cp命令可以一次复制多个文件，例如：
$cp *.txt *.doc *.bak /home
将当前目录中扩展名为txt、doc和bak的文件全部复制到/home目录中。
如果要复制整个目录及其所有子目录，可以用cp -R命令。

rm
rm命令用于删除文件或目录。
rm命令会强制删除文件，如果想要在删除时提示确认，可用rm -i命令。
如果要删除目录，可用rm -r命令。rm -r命令在删除目录时，每删除一个文件或目录都会显示提示，如果目录太大，响应每个提示是不现实的。这时可以用 rm -rf命令来强制删除目录，这样即使用了-i标志也当无效处理。

mv
mv命令用于移动文件和更名文件。例如：
$mv ye.txt /home
将当前目录下的ye.txt文件移动到/home目录下，
$mv ye.txt ye1.txt
将ye.txt文件改名为ye1.txt。
类似于跟cp命令，mv命令也可以一次移动多个文件，在此不再赘叙。

reboot
重启命令，不必多说。

halt
关机命令，不必多说。
<hr>

LINUX的目录树

/bin bin是binary的缩写。这个目录是对UNIX系统习惯的沿袭，存放着使用者最经常使用的命令。例如：cp,ls,cat。
/boot 这里存放的是启动LINUX时使用的一些核心文件。
/dev dev是device(设备)的缩写。这个目录下是所有LINUX的外部设备，其功能类似DOS下的.sys和Win下的.vxd。在LINUX中设备和文件是用同种方法访问的。例如:/dev/hda代表第一个物理IDE硬盘。
/etc 这个目录用来存放所有的系统管理所需要的配置文件和子目录。
/home 用户的主目录，比如说有个用户叫wang，那他的主目录就是/home/wang也可以用~wang表示，说到这里，打个岔，你现应该明白在我们访问一些个人网页(如http://www.netease.net/~zhangjia
如果这个网站的操作系统是LINUX，那就是表示/home/zhangjia。
/lib 这个目录里存放着系统最基本的动态链接共享库，其作用类似于Windows里的.dll文件。几乎所有的应用程序都需要用到这些共享库。
/lost+found 这个目录平时是空的，当系统不正常关机后，这里就成了一些无家可归的文件的避难所。对了，有点类似于DOS下的.chk文件。
/mnt 这个目录是空的，系统提供这个目录是让用户临时挂载别的文件系统。
/proc 这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。也就是说，这个目录的内容不在硬盘上而是在内存里啊！？
/root 系统管理员，也叫作超级权限者的用户主目录。当然系统的拥有者，总要有些特权啊！
/sbin s就是Super User的意思，也就是说这里存放的是一些系统管理员使用的系统管理程序。
/tmp 这个目录不用说，一定是用来存放一些临时文件的地方了。
/usr 这是个最庞大的目录，我们要用到的很多应用程序和文件几乎都存放在这个目录下。具体来说：
/usr/X11R6 存放X-Windows的目录；
/usr/bin 存放着许多应用程序；
/usr/sbin 给超级用户使用的一些管理程序就放在这；
/usr/doc 这就是LINUX文档的大本营；
/usr/include LINUX下开发和编译应用程序需要的头文件，在这找；
/usr/lib 存放一些常用的动态链接共享库和静态档案库；
/usr/local 这是提供给一般用户的/usr目录，在这安装软件最适合；
/usr/man man是什么，对了是帮助。这里就是帮助文档目录啊！
/usr/src LINUX开放的源代码，就存在这个目录，爱好者们别放过哦！
/var 这个目录中存放着那些不断在扩充着的东西，为了保持/usr的相对稳定，那些经常被修改的目录可以放在这个目录下，实际上许多系统管理员都是这样干的。顺带说一下系统的日志文件就在/var/log目录中。
这里列出了最常见的目录，根据LINUX发行套件的不同，目录结构也是有一定差别的。还有你自己建立的目录，我当然不知是存什么的。希望上面的文字能使你在大脑里留下了LINUX的全景图，以便大家深入掌握LINUX的其它方方面面。
<hr>

Linux各项系统开机服务的功能是什么

Linux各项系统开机服务的功能是什么

Linux在启动时要启动很多系统服务，它们向本地和网络用户提供了Linux的系统功能接口，直接面向应用程序和用户。但是，开启不必要或有漏洞的服务则会给操作系统带来安全和性能上的影响。下面我们以BluePoint Linux 2.0的开机服务为例，列表说明各项服务的功能（见附表）。

alsasound
Alsa声卡驱动程序支持。Alsa声卡驱动程序本来是为了 一种声卡Gravis UltraSound(GUS)而写的，该程序被证 明很优秀，于是作者就开始为一般的声卡写驱动程序。 Alsa和OSS/Free 及OSS/Linux兼容，但是有自己的接 口，甚至比OSS优秀。
amd
运行automount精灵程序，该精灵在必要时自动安装一些 本地设备和NFS文件系统。

apmd
apmd用来监视系统用电状态，并将相关信息通过syslogd 写入日志。也可以用来在电源不足时关机。

arpwatch
该程序主要用来维护以太网物理地址和IP地址的对应关系。

atalk
AppleTalk精灵程序。注意不要在后台运行该程序，该程 序的数据结构必须在运行其他进程前先花一定时间初始化。

atd
运行用户用At命令调度的任务。也在系统负荷比较低时 运行批处理任务。

autofs
当您需要时自动转载文件系统，而当您不需要时自动卸载。
bootparamd
该服务允许老的Sun工作站从Linux网络启动，它和rarp 现在很少使用，基本上被bootp和dhcp取代了。

crond
cron是Unix下的一个传统程序，该程序周期地运行用户 调度的任务。比起传统的Unix版本，Vixie版本添加了不 少属性，而且更安全，配置更简单。
dhcpd 该精灵提供了对动态主机控制协议(Dynamic Host Control Protocol)的访问支持。

gated
gated通过一个数据库提供了网络路由功能支持。它支持 各种路由协议，包括RIP版本1和2、DCN HELLO协议、 OSPF版本2以及EGP版本2到4。

gpm
gpm为文本模式下的Linux程序如mc(Midnight Commander)提供了鼠标的支持。它也支持控制台下鼠标 的拷贝，粘贴操作以及弹出式菜单。

httpd
http是著名的www服务器，可用来提供HTML文件以 及CGI动态内容服务。

inetd
因特网操作服务程序。监控网络对各种它管理的服务的需 求，并在必要的时候启动相应的服务程序。通常，inetd 管理的程序有telnet、ftp、rsh和rlogin。关闭inetd也就 关闭了这些由它管理的服务。

innd
inn是最流行的用户组新闻服务器。它允许您建立起本地 新闻服务器。配置有一定的难度，可以先阅读/usr/doc/ inn*文档获得帮助。
keytable 该程序的功能是转载您在/etc/sysconfig/keyboards里说 明的键盘映射表，该表可以通过kbdconfig工具进行选 择。您应该使该程序处于激活状态。

ldap
LDAP代表Lightweight Directory Access Protocol， 实现了目录访问协议的行业标准。

linuxconf
linuxconf是Linux下的一个有效的系统配置工具，该服 务允许远程运行。

lpd
lpd是系统打印守护程序，负责将lpr等程序提交给打印 作业。

mcserv
Midnight Commander服务进程允许远程机器上的用户 通过Midnight Commander文件管理器操作本机文件。服 务进程用PAM来验证用户，需要给出“用户名/口令” 以通过验证。

mysql 一个快速高效可靠的轻型SQL数据库引擎。

named 域名服务器，将Internet主机名解析为点分的IP地址。

netfs 负责装载/卸载NFS、Samba、NCP(Netware)文件系统。

network 激活/关闭启动时的各个网络接口。

nfs
NFS是一个流行的基于TCP/IP网络的文件共享协议。该 服务提供了NFS文件共享服务，具体的配置在/etc/ exports文件里。

nscd
该服务负责密码和组的查询，并且缓冲查询结果。如果您 的系统有比较慢的服务(如NIS和NIS+)，则应该启动该 服务。

pcmcia pcmcia主要用于支持笔记本电脑。

portmap portmap用来支持RPC连接，RPC被用于NFS以及NIS 等服务。

postgresql PostgreSQL关系数据库引擎。

proftpd proftpd是Unix下的一个配置灵活的ftp守护程序。

radvd 路由广播程序。

random
保存和恢复系统的高质量随机数生成器，这些随机数是系 统一些随机行为提供的。

routed
该守护程序支持RIP协议的自动IP路由表维护。RIP主要 使用在小型网络上，大一点的网络就需要复杂一点的协议。

rstatd Rstat协议允许网络上的用户获得同一网络上各机器的性能 参数。

rusersd 该服务使网络用户可以定位同一网络上的其他用户。

rwalld
Rwall协议允许远程用户向在同一系统中活跃着的终端发送 消息，类似wall的本地行为。

rwhod
允许远程用户获得运行rwho精灵的机器上所有已登录用户 的列表，与finger类似。

sendmail 大名鼎鼎的邮件服务器。

smb 启动和关闭smbd和nmbd精灵程序以提供SMB网络服务。

snmpd 简单网络管理协议(SNMP)的守护精灵。

syslog
syslog是操作系统提供的一种机制，守护程序通常使用这 种机制将各种信息写到各个系统日志文件。通常应该启动 该服务。

xfs X的字体服务器。
ypbind NIS/YP的客户端守护程序。如果您需要使用NIS/YP机 器，请启动这项服务，否则，关闭这项服务。

yppasswd
让NIS用户能够修改密码。运行在NIS域的服务器上。客 户端程序同样也叫yppasswd。

ypserv
标准NIS/YP网络协议的一个实现。允许主机名，用户名 和其他信息分布于网络各端。运行在NIS服务器上，客户 端不需要

Linux关机命令详解

Linux关机命令详解

在linux下一些常用的关机/重启命令有shutdown、halt、reboot、及init，它们都
可以达到重启系统的目的，但每个命令的内部工作过程是不同的，通过本文的介绍，希
望你可以更加灵活的运用各种关机命令。
1.shutdown
shutdown命令安全地将系统关机。
有些用户会使用直接断掉电源的方式来关闭linux，
这是十分危险的。因为linux与windows不同，其后台运行着许多进程，所以强制关机可能
会导致进程的数据丢失，使系统处于不稳定的状态，甚至在有的系统中会损坏硬件设备。
而在系统关机前使用shutdown命令，系统管理员会通知所有登录的用户系统将要关闭。
并且login指令会被冻结，即新的用户不能再登录。直接关机或者延迟一定的时间才关机
都是可能的，还可能重启。这是由所有进程〔process〕都会收到系统所送达的信号〔signal〕

决定的。这让像vi之类的程序有时间储存目前正在编辑的文档，而像处理邮件〔mail〕和
新闻〔news〕的程序则可以正常地离开等等。
shutdown执行它的工作是送信号〔signal〕给init程序，要求它改变runlevel。
Runlevel 0被用来停机〔halt〕，runlevel 6是用来重新激活〔reboot〕系统，
而runlevel 1则是被用来让系统进入管理工作可以进行的状态；这是预设的，假定没有-h也
没有-r参数给shutdown。要想了解在停机〔halt〕或者重新开机〔reboot〕过程中做了哪些
动作，你可以在这个文件/etc/inittab里看到这些runlevels相关的资料。
shutdown 参数说明:
[-t] 在改变到其它runlevel之前，告诉init多久以后关机。
[-r] 重启计算器。
[-k] 并不真正关机，只是送警告信号给每位登录者〔login〕。
[-h] 关机后关闭电源〔halt〕。
[-n] 不用init，而是自己来关机。不鼓励使用这个选项，而且该选项所产生的后果往
往不总是你所预期得到的。
[-c] cancel current process取消目前正在执行的关机程序。所以这个选项当然没有
时间参数，但是可以输入一个用来解释的讯息，而这信息将会送到每位使用者。
[-f] 在重启计算器〔reboot〕时忽略fsck。
[-F] 在重启计算器〔reboot〕时强迫fsck。
[-time] 设定关机〔shutdown〕前的时间。

2.halt----最简单的关机命令
其实halt就是调用shutdown -h。halt执行时，杀死应用进程，执行sync系统调用，
文件系统写操作完成后就会停止内核。
参数说明:
[-n] 防止sync系统调用，它用在用fsck修补根分区之后，以阻止内核用老版本的超

级块〔superblock〕覆盖修补过的超级块。
[-w] 并不是真正的重启或关机，只是写wtmp〔/var/log/wtmp〕纪录。
[-d] 不写wtmp纪录〔已包含在选项[-n]中〕。
[-f] 没有调用shutdown而强制关机或重启。
[-i] 关机〔或重启〕前，关掉所有的网络接口。
[-p] 该选项为缺省选项。就是关机时调用poweroff。

3.reboot
reboot的工作过程差不多跟halt一样，不过它是引发主机重启，而halt是关机。它
的参数与halt相差不多。
4.init
init是所有进程的祖先，它的进程号始终为1，所以发送TERM信号给init会终止所有的
用户进程、守护进程等。shutdown 就是使用这种机制。init定义了8个运行级别(runlevel)，
init 0为关机，init 1为重启。关于init可以长篇大论，这里就不再叙述。另外还有
telinit命令可以改变init的运行级别，比如，telinit -iS可使系统进入单用户模式，
并且得不到使用shutdown时的信息和等待时间。

上一页 目录 下一页

上一页 目录 下一页

linux技巧33条

1：HOST的乐趣：Host能够用来查询域名，然而它可以得到更多的信息。host -t mx l
inux.com可以查询出Linux.com的MX记录，以及处理Mail的Host的名字。Host -l linux
com会返回所有注册在linux.com下的域名。host -a linux.com则会显示这个主机的所
有域名信息。
2：找到命令需要的文件，也许你希望搞清楚一个命令它到底使用到了什么其他相关文件

和命令？试试ldd就可以了。如ldd traceroute，返回结果会是libc.so.6,lid-linux.s
o.2
3：定时！定时！你希望定时处理吗？你肯定希望能够让你轻松一些！crontab可以听候
您的指示。命令行中输入：crontab -e这进入文本编辑器，再在其中输入0 5 * * * wg
et -t3 -I/home/tom/URLs -N。这样每天早上5点就会运行这个wget命令。前五个是时间

参数：分 时 日 月 星期几。你可以 0 5 * * 6,0 command line here，这样就在每个
星期六和星期天5am运行。你需要man 5 crontab来得到更为详细的解释。随便说一下，
wget是一个取web页面的小程序。 console顶部显示现在时间：在/etc/bashrc或~/.bas
hrc中加入环境变量PROMPT_COMMAND。export PROMPT_COMMAND=echo -ne "33733[2;999
r33[1;1H33[00;44m33[K"`date`"33[00m338"
4：使用多达23个的虚拟终端（VT），假设你现在有6个，仿照/etc/inittab种的文件，
加入

8:234:respawn:/sbin/getty 38400 tty8
24:234:respawn:/sbin/getty 38400 tty24
要跳过VT7，因为X使用这个终端。然后telinit q以重新加载配置文件。使用左Alt+Fn在

1-6和8-12件切换，使用右Alt+Fn键在13-24之间切换。
5：Console切换：左Alt-方向键，循环切换console。效果和Alt+Fn一样。
6：简单的进行Dos转Unix：如果你的文本文件中每行结尾都有讨厌的Ctrl-M的话，使用这句命令能够快速的解决问题 sed s/^M// filename > newfilename
7：快速建立MP3播放列表：find /mnt/cdrom -name *.mp3 -print >> cd1.play.list. 这样就会把CDRom中的所有MP3文件建立一个名为cd1.play.list的文件。然后使用xmms或 x11amp就能进行播放了。
8：/proc中的重要信息
cat /proc/cpuinfo - CPU (i.e. vendor, Mhz, flags like mmx)
cat /proc/interrupts - 中断
cat /proc/ioports - 设备IO端口
cat /proc/meminfo - 内存信息(i.e. mem used, free, swap size)
cat /proc/partitions - 所有设备的所有分区
cat /proc/pci - PCI设备的信息
cat /proc/swaps - 所有Swap分区的信息
cat /proc/version - Linux的版本号
9：真正的Console界面：说实话无论是KDE和GNome我都没什么喜欢的，毕竟受Windows熏

陶了这么长时间了。但是在Console下其实才是真正的Cool界面。使用SVGATextMode可以

让你的命令行界面达到132x60(标准是80x25)。我的486（Trident9440）上使用的是132x43，那密密麻麻的字符真象Hollywood的几年前的电影（现在连电影里都是Windows了）

给人感觉神秘兮兮的。配置文件在/etc/TextConfig中，不过如果你的XWindow都无法正常显示的话，那估计这也用不了了（毕竟是好几年前的东西了）。
10：让你的硬盘飞起来: Linux下也可以使用32Bit I/O和DMA。使用/sbin/hdparm -c1 /dev/hda(hdb,hdc..)打开32Bit传输模式，使用命令 /sbin/hdparm -d1 /dev/hda(hdb,hdc...) 打开DMA。最后使用/sbin/hdparm -k1 /dev/hda 以使硬盘在Reset之后保持上

面的设定，这么一来，硬盘读写速度应该可以提高一倍以上。
11：微调你的Ext2: Linux下的命令tune2fs可以设定最大Mount次数(因为Linux即时在mounted clean状况下，达到一定的mount次数后，会强制e2fsck对于我这样的每天开关机

的来说很不方便)，tune2fs命令的其他用途请参见man。
12：禁止在Netscape中显示闪烁（Blink）Tag: ~/.Xdefaults中修改 Netscape*blinkingEnabled: False 之后运行 xrdb -Load ~/.Xdefaults
13：随时改变XTerm的字体大小:shift加+/-
14：Ext2的文件属性: lsattr列出属性，chattr可以改变属性。如 chattr +A filename 则系统不会修改其last access time属性；chattr +i 则可以避免被rm掉。
15：SIS6326显卡的配置：在Device中加入：DacSpeed 110 Option "no_bitblt" Option "sw_cursor"
16：使用strings可以把字符串从Binary文件中分离出来
17：你大概已经习惯了使用Top来观察系统使用情况，那就使一使使用Vmstat 1来看看c

pu的空闲程度（显示结果的最后一项）。
18：使用Watch命令，反复执行特定命令，如"watch ls -l /tmp/blah"，将一直观察文件的大小改变。
19：使用pppstats得到ppp连接的数据。
20：显示远程机器的NFS: showmount -e hostname
21：使用autofs免去你每次mount和umount之苦,再RedHat6.0中带了autofs的rpm,只要安

装好之后便可以享受自动加载文件系统的方便。 在/misc目录下，创建一个cd目录，如果使用其自带的auto.misc和auto.master的话，马上就可以实现自动加载功能了。
22：在Bash Shell中，可以用！重复上几次输入过的命令，这样就可以省去往上滚动的麻烦了。使用ctl+R可以互动的检索以前使用过的命令行。
23：使用cd -在上两个cd /directory/操作中切换。如先是cd /home/foo接着是 cd /home/maison则接着打cd - 会进入 /home/foo再打cd - 会进入 /home/maison。
24：Bash中的快捷键：ESCAPE-B 会向前条一个单词，Ctrl-K会删除从光标到行末所有字

符。
25：在PS命令中使用grep，例：ps auxw | grep gpm 会列出所有含有gpm字符的进程。

26：在同一个命令行上同时执行多条命令：用分号分隔。如编译一个源文件：type ./configure ; make ; make install 。之后你就可以去散步了（不要在电脑前花费了太多

时间了，生活中还有很多美好的东西。）
27：把运行结果输出到一个文件中，如 ls -l > foo.file。
28：使用Top命令监视系统状况，内存使用情况，开机时间，进程状况，每个用户的使用

情况等等。使用S命令可以改变其刷新的时间：如.5将会使其每半秒钟更新一次。
29：同时Unzip多个文件,unzip *不正确，而是unzip "*"。
30：建立符号连结，这大概是最有用的功能了，ln -s real-file symbolic-file，无论

是设备，文档还是脚本程序。
31：改变显示的单位,如du -hm 将以M为单位输出结果，同样的du -hk则以K为单位输出
。
32：自动Logoff,在~/.bashrc中加入这么一句话：TMOUT=300则会在5分钟之后logoff,不

过如果less这类程序运行则不会启动。
33：在使用less 或 more时，直接启动编辑器：在less中打入v键，会立即启动vi或其他

你在环境变量中指明的编辑器，但是more只能使用vi。

上一页 目录 下一页

上一页 目录 下一页

Linux使用的一些基本问题(新手必读)

新手入门A 在linux下如何访问windows的分区。
#mkdir /mnt/win
创建一个加载点目录win,象cdrom、floppy的加载点目录都在/mnt下.
#mount -t vfat /dev/hda? /mnt/win
/dev/hda?可用#fdisk -l查看对应的windows
分区,进行mount.
-t
选项是给出分区格式.
-o
如果想正确的显示中文文件名可使用-o iocharset=cp936选项.
/etc/fstab文件是启动时linux加载的一些文件系统，你可以把它也写进去：
e.g.
/dev/hda? /mnt/win vfat defaults,iocharset=cp936 0 0

B 在/etc/inittab文件中有一行id:5:initdefault
5：以图形方式登陆。
3: 以文本方式登陆。
更详细的解释可以读读此文件.

C 用Xconfigurator命令可以配置X window，更改分辨率等..也可以setup命令有许多系统配置选项.(还有快捷键: ctrl+alt+"+" 和ctrl+alt+"-")

D 切换WM(window managers)窗口管理器可以用switchdesktop命令.
#switchdesk gnome OR #switchdesk kde,在X中的虚拟终端可以用#switchdesk在图形界面中选择.
还有一种方法就是在自己的$HOME(#echo $HOME)目录下创建.xinitrc文件.
#cd
#vi .xinitrc
exec gnome-session

: x
按esc键，输入": x"保存退出.

E filename.tar.bz2文件使用tar xjvf filename.tar.bz2命令解开.
tar.gz打包程序的一般安装方法:
#cd /tmp
#tar xvzf .../file.tar.gz
#cd file
#./configure
#make
#make install
具体可看看它的Readme or INSTALL
对于rpm打包的程序用rpm -ivh file.rpm进行安装。-e卸载.
升级用-Uvh选项。-Fvh更新，只有系统中存在rpm的旧就版本，才会升级安装。-Uvh选项是不管系统中有没有旧版本，都会安装。
查询用-q选项,-V是验证选项，都可综合下面的选项使用：
-l列出安装文件列表。

-i软件包信息。
-p软件包。
-f文件。
-d列出文档文件。
如：
rpm -q mozilla;rpm -ql mozilla;rpm -qi mozilla
rpm -qpl mozilla...rpm;rpm -qpi mozilla..rpm
以上命令要保证当前目录有mozilla...rpm.
rpm -qf /usr/bin/mozilla
看此文件属于那个软件包。
rpm -Va验证所有的软件包；rpm -Vf /bin/vi;rpm -Vp mozilla...rpm;rpm -V mozilla.
......

G 你可以用下列三种方法获得某个命令的帮助。
#command --help
this is simply.
#info command
#man command

H 在安装某个rpm包时可能遇到依赖性问题。
例如：
#cd /mnt/cdrom/RedHat/RPMS
#rpm -ivh kdemultimedia-2.2-4
error: faild dependencies
timidity++ is needed by kdemultimedia-2.2-4
libzrtskde.so....
libDCOP.so.1...
.....
我并不推荐用--nodeps选项忽略依赖性问题进行安装，可能不能运行。
要先将这些文件装上，它是属于那个rpm包呢？
可以去rpmfind.net搜索确定rpm包，下载安装。
或者在CDROM中搜索安装，下面介绍一个shell程序，帮你搜索包含依赖文件的rpm包。

```
for pkg in *.rpm
do
rpm -pql $pkg|grep $1>/dev/null && echo $pkg
done
```

将其写入ffrpm
#chmod u+x ffrpm
#cp ffrpm /bin
#cd /mnt/cdrom/Redhat/RPMS
#ffrpm findfile
具体请看此链接:
http://www.linuxforum.net/forum/showflat.php?Cat=&Board=newbie&Number=141576&page=&view=&sb=&o=&vc=1

G升级内核的方法，感谢ccie!

在make menuconfig之前，到/usr/include下
#rm -fr scsi asm linux
#ln -s /usr/src/linux/include/linux linux
#ln -s /usr/src/linux/include/asm-i386 asm
#ln -s /usr/src/linux/include/scsi scsi
这里有三种方法配置内核#make config、#make menuconfig、#make xconfig——这个在X下运行。
然后，make dep,make clean,make bzImage,make modules,make modules_install
再把/usr/src/linux/System.map拷贝到/boot下，把bzImage拷贝到/boot下。修改/etc/lilo.conf(如果用lilo的话），运行/sbin/lilo -v.

H 配置鼠标使用mouseconfig命令，使用滚轮修改下面两个文件都可以:
/etc/sysconfig/mouse
将其改成：
FULLNAME="Microsoft - IntelliMouse (PS/2)"
MOUSETYPE="imps2"
XEMU3="no"
XMOUSETYPE="IMPS/2"
感谢gywx
或者修改/etc/X11/XF86Config-4
Section "InputDevice"
Identifier "Mouse0"
Driver "mouse"
Option "Protocol" "IMPS/2"
Option "Device" "/dev/psaux"
Option "ZAxisMapping" "4 5"
Option "Emulate3Buttons" "no"
EndSection
也可以在mouseconfig中选鼠标型号选择microsoft的智能鼠或logitech的mouseman即可.

J 每次修改/etc/lilo.conf文件后要运行lilo命令，将其写入mbr.
比如你要缺省启动win:
prompt
timeout=50
default=winme
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
linear
image=/boot/vmlinuz-2.4.7-10
label=linux
initrd=/boot/initrd-2.4.7-10.img
read-only
root=/dev/hda7
other=/dev/hda1
optional
label=winme

K 进入X，如果屏幕歪掉啦，当然可以直接调显示器，但可能跟win冲突，也可以#xvidtune调整，调整好后记下如下的一些参数:

HDisplay:1024 HSyncStart:1098 HSyncEnd:1188 HTotal:1376
VDisplay:768 VSyncStart:769 VSyncEnd:772 VTotal:808
Pixel Clock(MHz):94.50
Horizontal Sync(KHz):68.68
Vertical Sync(Hz):85.00

修改/etc/X11/XF86Config-4文件如下：

```
Section "Monitor"
Identifier "___A770"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 68.68
VertRefresh 85.00
Option "dpms"
# 1024x768 @ 85 Hz, 68.68 kHz hsync /*这句是注释，可以没有*/
Modeline "1024x768" 94.50 1024 1092 1188 1376 768 769 772 808 +HSync +VSync
EndSection
```

L 重装win会导致lilo、grub的无效， 无法从MBR中引导linux,这时有一张启动盘就很重要啦，你可以进入linux,对于lilo,输入#lilo -v -v -v,对于grub,输入#grub-install /dev/hda,即可。

上一页 目录 下一页

上一页 目录 下一页

LINUX文件格式简介

⊙ *.001, *.002 ....

大致上有以下几种可能，必须先分辨是哪种，再用对应的程序解：

1. ARJ 可以指定而造出 .001, .002 的文件
用 arj 解
2. DOS backup 造出的文件是 .001 .002
用 restore 解
3. AIN 分片压缩的档名是 .AIN, .001, .002
用 AIN 解

⊙ *.aif

MAC 上常用的音效档。PC上可用 plany 播放

⊙ *.arc

压缩档，可用 pkunpak 解压及 pak.exe 来解压；

⊙ *.arj, *.a01, *.a02 ...
这是一种压缩档，用arj 可以压缩并解压。
*.a01 等是使用arj 的分片功能时分出来的文件。

⊙ *.au
SUN 的音效档，在PC 上可用 plany 放出

⊙ *.avi
avi 是microsoft 定的一种动画档，其标准为影像加声音。
Win95 下可直接执行媒体播放程序播放。

⊙ *.bat
DOS 的 batch档，其内容为一般的文字档

⊙ *.bmp (Windows Bitmap Format)
图形档，大部份秀图程序都有支持

⊙ *.cap
画面狩猎者的输出格式，可以在原版画面狩猎者的说明书
中找到完整说明。

⊙ *.cdr (Corel DRAW)
Corel DRAW图形档

⊙ *.cmf (Creative Music File)
Creative 公司所使用类似MIDI的格式。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/cmfved10.zip

⊙ *.com
DOS 的可执行档

⊙ *.cru (crush)
压缩格式。

⊙ *.dbf (dBase file)
可到 simtel 的 dbase 目录找到相关软体。

⊙ *.dcf (Disk Copy Fast)
Disk Copy Fast造出的Image File! 可用dcf来解!!

⊙ *.ddi

diskdupe 做出来的 disk image 档
可用 x2file 来解。

⊙ *.dib
图形格式，类似 *.bmp，通常可以处理 *.bmp 的程序都可以用来处理
*.dib。

⊙ *.dl
一种动画格式，可以用 dl-view 来看。
ftp://softu1.ncu.edu.tw/pub5/News/oldexe/dl-view.exe

⊙ *.dsk
diskrw 造出来的 disk image 档。
ftp://ftp.fcu.edu.tw/pub4/CDROM/simtel2/disc2/diskutil/dskrw237.zip

⊙ *.dxf (Autodesk Drawing Exchange Format)
AutoCAD 2-d 图形档

⊙ *.eps
Encapsulated PostScriptd

⊙ *.exe
DOS 的可执行档

⊙ *.F
一种压缩格式!
ftp://ftp.edu.tw/USENET/comp.sources.misc/volume35/freeze

⊙ *.fli, *.flc
一种动画格式，*.fli 规格是小於 320*200*256色的，*.flc为
fli 的扩充规格，在dos 下可以用 aaplay , display 播放

⊙ *.gif (Graphics Interchange Format)
CompuServe 定图形档，大部份秀图程序都有支持

⊙ *.gl
一种动画格式，可以用 grasp4 来看。
ftp://ftp.edu.tw/PC/garbo/animation/grasp4.zoo

⊙ *.gz

压缩档，在unix 上可用 "gzip -d 档名" 解开

ftp://ftp.edu.tw/Chinese/ifcss/software/dos/utils/gzip124.exe(for dos)

⊙ *.ha

压缩档。DOS 下用 HA 来压缩/解压。

ftp://ftp.edu.tw/PC/garbo/arcers/ha098.zip

⊙ *.hap

压缩档，压缩率稍比 zip 好一点，目前只有 dos 的程序。

ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/hap305bp.zip

⊙ *.hpk

压缩档。DOS下用 hpack 来压缩/解压。

ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/hpack78.zip

⊙ *.hqx

.hqx是MAC 上一种编码格式，主要是把8bits 的文件编为7bits，以利传输。在PC 上可以用BINHEX 解开。

ftp://ftp.edu.tw/PC/Simtel/msdos/mac/binhex13.zip

⊙ *.html, *.htm (Hyper Text Markup Language)

WWW 用的标准格式，在PC 上可以利用底下几个程序来读

ftp://ftp.edu.tw/WWW/client/doslynx(for dos, 只有文字，没有图形)

ftp://ftp.edu.tw/WWW/netscape/navigator/2.02/windows/n16e202.exe

ftp://ftp.edu.tw/WWW/netscape/navigator/2.02/windows/n32e202.exe

⊙ *.ico

Windows Icon files

⊙ *.iff

图形档，Amiga Images, Deluxe Paint 使用

⊙ *.img

① hd-copy 做出来的 disk image 档。

ftp://ftp.cis.nctu.edu.tw/Msdos/diskutil/hd-cp20a.zip

② 图形档 (GEM Images)

⊙ *.jis
JIS (ジロス) = Japanese Industrial Standard 日本工业规格
JIS 在这指的应该是日本用的一种文字内码，一般的 *.jis
都是文字档，可以用南极星来编修。
ftp://ftp.edu.tw/PC/Simtel/msdos/editor/njj30*

⊙ *.jpg
图形档(JFIF)，可以用qpeg,display 来看

⊙ *.lbm
图形档， Deluxe Paint使用，编码方式类似 *.iff

⊙ *.lzh
压缩档，可用 lha 压缩及解压
ftp://ftp.edu.tw/PC/garbo/arcers/lha255e.exe

⊙ *.mag
日本常用的一种图形档。 PC 上可以用 gv,display 来看
ftp://ftp.edu.tw/PC/garbo/graphics/disp189a.zip
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/gv057.zip

⊙ *.mid
midi 音效档

⊙ *.mod
类似midi 音效档。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/mp219b.zip

⊙ *.mov
Quick Time 的文件，一种动画格式，为Apple 公司规画，在PC 上
可以找 Quick Time for Windows (qtw) 来看!
ftp://ftp.tem.nctu.edu.tw/Windows/multimed/qtw203.exe
ftp://ftp.cis.nctu.edu.tw/OS/Windows/95/multimedia/graphic/qt32inst.exe

⊙ *.mp2, *.mpa (Mpeg Audio Layer 2)
一般可以放mpeg 文件的player 大多可以播放。
Dos:
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/ampeg43.zip

(可在.wav,.pcm<->.mp2转换)
Windows:
ftp://ftp.cis.nctu.edu.tw/Windows/3.x/graphics/softpeg13.zip
ftp://ftp.cis.nctu.edu.tw/Windows/95/multimedia/graphic/mpw32-5n.zip

⊙ *.mp3 (Mpeg Audio Layer 3)
Windows:
ftp://ftp.thu.edu.tw/upload/win95/winplay3.zip

⊙ *.mpg
一种动画格式，包含声音及影像，包含声音的方法有两种，一种是
直接放在 .mpg里，一种是另外用 .wav档储存。
ftp://ftp.edu.tw/mirror/Cica/win3/desktop/vmpeg12a.zip(For Dos)
Windows:
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/softpeg2.zip
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/xing21.zip

⊙ *.msl
一种动画格式，内含声音影像。
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/playmsl.exe(Dos/V)
ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/Maslv41.lzh(Win3.1)

ftp://ftp.edu.tw/Graphics/Anime/Software/ImageViewer/Masl32.lzh(Win95)

⊙ *.pak
压缩档, 由 pak.exe 产生、解压。
ftp://ftp.edu.tw/PC/garbo/arcers/pak251.exe

⊙ *.pcc
图形档, 参见 *.pcx

⊙ *.pcd
图形档(Kodak PhotoCD)

⊙ *.pcx (ZSoft Paint)
图形档，大部份秀图程序都有支持

⊙ *.pdf (Portable Document Format)
Adobe 制订的文件档。可用 acroread.exe 来读取。
ftp://ftp.tku.edu.tw/pc/winsock/WWW/Plug-in/AcrobatAmber

⊙ *.pic
图形档 (Lotus 1-2-3 Pictures)

⊙ *.png (Portable Network Graphics)
图形档。为取代 GIF 而制订的图形档格式。
QPV, display, gwswin 等软件都有支持。

⊙ *.ppm (Portable Pixmap)
图形档。

⊙ *.ps
post script，一种page description language，如果你使用的列表机
有支持PS 格式的话，直接列印就好了。不然PC 上可以抓ghostview,
ghostscript 等东西来用。
ftp://ftp.edu.tw/PC/Simtel/msdos/postscrp/gs*for dos
ftp://ftp.cis.nctu.edu.tw/OS/Windows/misc/postscript/*for win
ftp://ftp.edu.tw/Windows/Windows95/Cica/miscutil/rops3244.zip

⊙ *.psd
PhotoShop 标准档。Shareware 中 Paint Shop Pro 可以读非
Multi-layer的*.psd。
ftp://ftp.cis.nctu.edu.tw/OS/Windows/3.x/graphics/psp311.zip

⊙ *.rar
压缩档。可以用 rar 来压缩解压。
ftp://ftp.cis.nctu.edu.tw//Msdos/archiver/rar/rar200/rar200.exe

⊙ *.ras (Sun Raster files)
图形档

⊙ *.raw
图形档(Raw GrayScale)

⊙ *.rcp
一种 Midi 音效档

⊙ *.rle
压缩过的 *.bmp(Run Length Compressed)

⊙ *.rtf
Rich Text Format

⊙ *.s3m
S3M 是由 Future Crew 所设计的一种音乐档格式，可用 iplay 来播放。
ftp://ftp.edu.tw/PC/garbo/sound/iplay121.zip

⊙ *.sdn (Software Distribution Network 组织)
以前是采用 pak.exe 来压缩, 自 1994 以後改用 arj.exe 来压。

⊙ *.sit
Mac 上的 Stuff It 压出来的东西，压缩采用的方法是把文件集中起
来，然後用 RLE, Huffman, LZW 等方法压缩起来。底下是两个 PC
上的解压程序。
ftp://ftp.edu.tw/PC/garbo/arcers/unsit30.zip
ftp://ftp.edu.tw/PC/garbo/arcers/unstufit.zoo

⊙ *.snd
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.sqz
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/sqz1083e.exe

⊙ *.tar
为了备份方便而把所有文件放在一起的文件，在unix 上可直接下指
令`tar -xf 档名` 来解开
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/tar4dos.zip

⊙ *.tar.gz
先解开 *.gz 再把解开的 *.tar 解开。
如果是用 GNU 的tar 的话可直接用 "tar -zxvf *.tgz" 来解开。

⊙ *.td0
*.td0 是由 teledisk 造出来的disk image 档。

ftp://ftp.tem.nctu.edu.tw/Msdos/arcers/teled215.zip

⊙ *.tga (Truevision (Targa) File Format)
图形档，svga, vpic, cshow, qpeg 都有支持

⊙ *.tgz
.tgz就是.tar.gz，先解 *.gz 再把解开的 *.tar 解开。
如果是用 GNU 的tar 的话可直接用 "tar -zxvf *.tgz" 来解开。

⊙ *.tif, *.tiff (Tag Image File Format)
图形档， cshow、display 及一些影像绘图软件都有支持。

⊙ *.ttf
Windows True Type Font

⊙ *.u01 *.u02 .....
这是 unpack 造出来的，用 pack 可以合起来。
unpack 是把一个大文件分成数个小档的程序。
ftp://ftp.edu.tw/Chinese/Taiwan-netinfo/SEEDNET/success/apart/pack.exe

⊙ *.uc2
压缩档(UltraCompressor II)。PC 上用 uc2 压缩/解压。
ftp://ftp.edu.tw/PC/Simtel/msdos/archiver/uc2r3.exe

⊙ *.uue, *.uud
可能是由 uuencode 造出的文件，在unix 上可用 uudecode
来解码，在dos 上可以抓底下的程序来用。
ftp://ftp.edu.tw/PC/garbo/decode/uuexe651.zip

⊙ *.voc
一种音效档，可用 plany 放。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.wav
一种音效档，可用plany 放出。
ftp://ftp.edu.tw/PC/Simtel/msdos/sound/plany15.zip(dos)
ftp://ftp.edu.tw/PC/ncsa/Windows/Mosaic/viewers/wplny12a.zip(win)

⊙ *.win
图形档，类似 *.tga

⊙ *.wmf
Windows Metafiles

⊙ *.xbm (X BitMap)
ftp://ftp.edu.tw/PC/garbo/graphics/disp189a.zip

⊙ *.xmi
音效档。
ftp://ftp.cis.nctu.edu.tw/Ultrasound/submit/xmi2mid.zip

⊙ *.z
参见 *.gz

⊙ *.Z
压缩档，在unix 上用 uncompress 解开

⊙ *.zip
这是一种压缩档，可以用 pkunzip 来解压，用 pkzip 可以造出这种档。
ftp://ftp.edu.tw/PC/Simtel/msdos/zip/pkz204g.exe

⊙ *.zoo
这是一种压缩档。由 zoo 这个程序来处理。
ftp://ftp.edu.tw/PC/Simtel/msdos/zoo/zoo210.exe

上一页 目录 下一页

上一页 目录 下一页

Linux系统命令(网络通信)

ytalk

功能说明：与其他用户交谈。

语　　法：ytalk [-isxY][-h<主机名称IP地址>][用户名称...]

补充说明：通过ytalk指令，你可以和其他用户线上交谈，如果想和其他主机的用户交谈，在用户名称后加上其主机名称或IP地址即可。

参　　数：
-h<主机名称IP地址> 指定交谈对象所在的远端主机。
-i 用提醒声响代替显示信息。
-s 在指令提示符号先开启ytalk交谈窗。
-x 关闭图形界面。
-Y 所有必须回应yes或no的问题，都必须用大写英文字母"Y"或"N"回答。

----

write

功能说明：传送信息。

语　　法：write [用户名称][终端机编号]

补充说明：通过write指令可传递信息给另一位登入系统的用户，当输入完毕后，键入EOF表示信息结束，write指令就会将信息传给对方。如果接收信息的用户不只登入本地主机一次，你可以指定接收信息的终端机编号。

----

wall(write all)

功能说明：传送信息。

语　　法：wall [公告信息]

补充说明：通过wall指令可将信息发送给每位同意接收公众信息的终端机用户，若不给予其信息内容，则wall指令会从标准输入设备读取数据，然后再把所得到的数据传送给所有终端机用户。

----

uux

功能说明：在远端的UUCP主机上执行指令。

语　　法：uux [-bcCIjlnrvz][-a〈地址〉][-g〈等级〉][-s〈文件〉][-x〈层级〉][--help][指令]

补充说明：uux可在远端的UUCP主机上执行指令或是执行本机上的指令，但在执行时会使用远端电脑的文件。

参　　数：
-或-p或--stdin 直接从键盘读取要执行的指令。
-a〈地址〉或--requestor〈地址〉 执行邮件地址，以便寄送状态信息。
-b或--erturn-stdin 在屏幕上显示状态信息。
-c或--nocopy 不用将文件复制到缓冲区。
-C或--copy 将文件复制到缓冲区。
-g〈等级〉或--grade〈等级〉 指定文件传送作业的优先顺序。
-I或--config file 指定uux配置文件。
-j或--jobid 显示作业编号。
-l或--link 将本机上的文件连接到缓冲区。
-n或--notification=no 无论发生任何状态，都不寄邮件通知用户。
-r或--nouucico 不要立即启动uucico服务程序，仅将作业送到队列中，然后再执行。
-s〈文件〉或--status〈文件〉 将完成状态保存为指定的文件。
-v或--version 显示版本信息。
-x〈层级〉或--debug〈层级〉 指定排错层级。
-z或--notification=error 若发生错误，则以邮件来通知用户。
--help 显示帮助。

----

uustat

功能说明：显示UUCP目前的状况。

语　　法：uustat [-aeiKmMNpqQRv][-B〈行数〉][-c〈指令〉][-C〈指令〉][-I〈配置文件〉][-k〈工作〉][-o〈小时〉][-r〈工作〉][-s〈主机〉][-S〈主机〉][-u〈用户〉][-U〈用户〉][-W〈附注〉][-y〈小时〉][-x〈层级〉][--help]

补充说明：执行uucp与uux指令后，会先将工作送到队列，再由uucico来执行工作。uustat可显示，删除或启动队列中等待执行的工作。

参　　数：
-a或-all 显示全部的UUCP工作。
-B〈行数〉或--mail-lines〈行数〉 与-M或-N参数一并使用，用来指定邮件中要包含多少行的信息。
-c〈指令〉或--command〈指令〉 显示与〈指令〉有关的工作。
-C〈指令〉或--not-command〈指令〉 显示与〈指令〉无关的工作。
-e或--executions 仅显示待执行的工作。

-i或--prompt 针对队列中的每项工作，询问使用是否要删除工作。
-I<配置文件>或--config<配置文件> 指定配置文件。
-k<工作>或--kill<工作> 删除指定的工作。
-m或--status 删除全部的工作。
-M或-mail 将状态信息邮寄给UUCP管理员。
-N或--notify 将状态信息分别邮寄给提出该项工作的用户。
-o<小时>或--older-than<小时> 显示超过指定时数的工作。
-p或--ps 显示负责UUCP锁定的程序。
-q或--list 显示每台远端主机上所要执行工作的状态。
-Q或--no-list 不显示工作。
-r<工作>或--rejuvenate<工作> 重新启动指定的工作。
-R或--rejuvenate-all 重新启动全部的工作。
-s<主机>或--system<主机> 显示与<主机>有关的工作。
-S<主机>或--not-system<主机> 显示与<主机>无关的工作。
-v或--version 显示版本信息。
-u<用户>或--user<用户> 显示与<用户>有关的工作。
-U<用户>或--not-user<用户> 显示与<用户>无关的工作。
-W<附注>或--comment<附注> 要放在邮件信息中的附注。
-y<小时>或--younger-than<小时> 显示低于指定时数的工作。
-x<层级>或--debug<层级> 指定排错层级。
--help 显示帮助。

----

uuname

功能说明：显示全部的UUCP远端主机。

语　　法：uuname [-alv][-I<配置文件>][--help]

补充说明：uuname可显示UUCP远端主机。

参　　数：
-a或--aliases 显示别名。
-I<配置文件>或--config<配置文件> 指定程序的配置文件。
-l或--local 显示本机名称。
-v或--version 显示版本信息。
--help 显示帮助。

----

uulog

功能说明：显示UUCP记录文件。

语　　法：uulog [-DFISv][-<行数>][-f<主机>][-I<配置文件>][-n<行数>][-s<主机>][-u<用户>][-X<层级>][--help]

补充说明：uulog可用来显示UUCP记录文件中记录。

参　　数：
-D或--debuglog 显示排错记录。
-f<主机>或--follow<主机> 与-F参数类似，但仅显示与指定主机相关的记录。
-I<配置文件>或--config<配置文件> 指定程序的配置文件。
-<行数>,-n<行数>或--lines<行数> 显示记录文件中，从最后算起指定行数的数值。
-s<主机> 仅显示记录文件中，与指定文件相关的记录。
-S或--statslog 显示统计记录。
-u<用户>或--suer<用户> 仅显示记录文件中，与指定用户相关的记录。
-v或--version 显示版本信息。
-X<层级>或--debug<层级> 设定排错层级。
--help 显示帮助。

----

tty(teletypewriter)

功能说明：显示终端机连接标准输入设备的文件名称。

语　　法：tty [-s][--help][--version]

补充说明：在Linux操作系统中，所有外围设备都有其名称与代号，这些名称代号以特殊文件的类型存放于/dev目录下。你可以执行tty指令查询目前使用的终端机的文件名称。

参　　数：
-s或--silent或--quiet 不显示任何信息，只回传状态代码。
--help 在线帮助。
--version 显示版本信息。

----

traceroute

功能说明：显示数据包到主机间的路径。

语　　法：traceroute [-dFlnrvx][-f<存活数值>][-g<网关>...][-i<网络界面>][-m<存活数值>][-p<通信端口>][-s<来源地址>][-t<服务类型>][-w<超时秒数>][主机名称或IP地址][数据包大小]

补充说明：traceroute指令让你追踪网络数据包的路由途径，预设数据包大小是40Bytes，用户可另行设置。

参　　数：
-d 使用Socket层级的排错功能。
-f<存活数值> 设置第一个检测数据包的存活数值TTL的大小。
-F 设置勿离断位。
-g<网关> 设置来源路由网关，最多可设置8个。
-i<网络界面> 使用指定的网络界面送出数据包。
-I 使用ICMP回应取代UDP资料信息。
-m<存活数值> 设置检测数据包的最大存活数值TTL的大小。
-n 直接使用IP地址而非主机名称。
-p<通信端口> 设置UDP传输协议的通信端口。
-r 忽略普通的Routing Table，直接将数据包送到远端主机上。
-s<来源地址> 设置本地主机送出数据包的IP地址。
-t<服务类型> 设置检测数据包的TOS数值。
-v 详细显示指令的执行过程。
-w<超时秒数> 设置等待远端主机回报的时间。
-x 开启或关闭数据包的正确性检验。

----

testparm(test parameter)

功能说明：测试Samba的设置是否正确无误。

语　　法：testparm [-s][配置文件][<主机名称>]

补充说明：执行testparm指令可以简单测试Samba的配置文件，假如测试结果无误，Samba常驻服务就能正确载入该设置值，但并不保证其后的操作如预期般一切正常。

参　　数：
-s 不显示提示符号等待用户按下Enter键，就直接列出Samba服务定义信息。

----

telnet

功能说明：远端登入。

语　　法：telnet [-8acdEfFKLrx][-b<主机别名>][-e<脱离字符>][-k<域名>][-l<用户名称>][-n<记录文件>][-S<服务类型>][-X<认证形态>][主机名称或IP地址<通信端口>]

补充说明：执行telnet指令开启终端机阶段作业，并登入远端主机。

参　　数：
-8 允许使用8位字符资料，包括输入与输出。
-a 尝试自动登入远端系统。
-b<主机别名> 使用别名指定远端主机名称。
-c 不读取用户专属目录里的.telnetrc文件。
-d 启动排错模式。
-e<脱离字符> 设置脱离字符。
-E 滤除脱离字符。
-f 此参数的效果和指定"-F"参数相同。
-F 使用Kerberos V5认证时，加上此参数可把本地主机的认证数据上传到远端主机。
-k<域名> 使用Kerberos认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名。
-K 不自动登入远端主机。
-l<用户名称> 指定要登入远端主机的用户名称。
-L 允许输出8位字符资料。
-n<记录文件> 指定文件记录相关信息。
-r 使用类似rlogin指令的用户界面。
-S<服务类型> 设置telnet连线所需的IP TOS信息。
-x 假设主机有支持数据加密的功能，就使用它。
-X<认证形态> 关闭指定的认证形态。

----

tcpdump

功能说明：倾倒网络传输数据。

语　　法：tcpdump [-adeflnNOpqStvx][-c<数据包数目>][-dd][-ddd][-F<表达文件>][-i<网络界面>][-r<数据包文件>][-s<数据包大小>][-tt][-T<数据包类型>][-vv][-w<数据包文件>][输出数据栏位]

补充说明：执行tcpdump指令可列出经过指定网络界面的数据包文件头，在Linux操作系统中，你必须是系统管理员。

参　　数：
-a 尝试将网络和广播地址转换成名称。
-c<数据包数目> 收到指定的数据包数目后，就停止进行倾倒操作。
-d 把编译过的数据包编码转换成可阅读的格式，并倾倒到标准输出。
-dd 把编译过的数据包编码转换成C语言的格式，并倾倒到标准输出。
-ddd 把编译过的数据包编码转换成十进制数字的格式，并倾倒到标准输出。
-e 在每列倾倒资料上显示连接层级的文件头。
-f 用数字显示网际网络地址。
-F<表达文件> 指定内含表达方式的文件。
-i<网络界面> 使用指定的网络截面送出数据包。
-l 使用标准输出列的缓冲区。

-n 不把主机的网络地址转换成名字。
-N 不列出域名。
-O 不将数据包编码最佳化。
-p 不让网络界面进入混杂模式。
-q 快速输出，仅列出少数的传输协议信息。
-r<数据包文件> 从指定的文件读取数据包数据。
-s<数据包大小> 设置每个数据包的大小。
-S 用绝对而非相对数值列出TCP关联数。
-t 在每列倾倒资料上不显示时间戳记。
-tt 在每列倾倒资料上显示未经格式化的时间戳记。
-T<数据包类型> 强制将表达方式所指定的数据包转译成设置的数据包类型。
-v 详细显示指令执行过程。
-vv 更详细显示指令执行过程。
-x 用十六进制字码列出数据包资料。
-w<数据包文件> 把数据包数据写入指定的文件。

----

talk

功能说明：与其他用户交谈。

语　　法：talk [用户名称][终端机编号]

补充说明：通过talk指令，你可以和另一个用户线上交谈。

----

statserial(status ofserial port)

功能说明：显示串口状态。

语　　法：statserial [-dnx][串口设备名称]

补充说明：statserial可显示各个接脚的状态，常用来判断串口是否正常。

参　　数：
-d 以10进制数字来表示串口的状态。
-n 仅显示一次串口的状态后即结束程序。
-x 与-n参数类似，但是以16进制来表示。

----

smbd(samba daemon)

功能说明：Samba服务器程序。

语　　法：smbd [-aDhoP][-d<排错层级>][-i<范围>][-l<记录文件>][-O<连接槽选项>][-p<连接端口编号>][-s<配置文件>]

补充说明：smbd为Samba服务器程序，可分享文件与打印机等网络资源供Windows相关的用户端程序存取。

参　　数：
-a 所有的连线记录都会加到记录文件中。
-d<排错层级> 指定记录文件所记载事件的详细程度。
-D 使用此参数时，smbd会以服务程序的方式在后台执行。
-h 显示帮助。
-i<范围> 指定NetBIOS名称的范围。
-l<记录文件> 指定记录文件的名称。
-o 每次启动时，会覆盖原有的记录文件。
-O<连接槽选项> 设置连接槽选项。
-p<连接端口编号> 设置连接端口编号。
-P 仅用来测试smbd程序的正确性。
-s<配置文件> 指定smbd的设置文件。

----

smbclient(samba client)

功能说明：可存取SMB/CIFS服务器的用户端程序。

语　　法：smbclient [网络资源][密码][-EhLN][-B][-d<排错层级>][-i<范围>][-I][-l<记录文件>][-M][-n][-O<连接槽选项>][-p][-R<名称解析顺序>][-s<目录>][-t<服务器字码>][-T][-U<用户名称>][-W<工作群组>]

补充说明：SMB与CIFS为服务器通信协议，常用于Windows95/98/NT等系统。smbclient可让Linux系统存取Windows系统所分享的资源。

参　　数：
[网络资源] [网络资源]的格式为//服务器名称/资源分享名称。
[密码] 输入存取网络资源所需的密码。
-B 传送广播数据包时所用的IP地址。
-d<排错层级> 指定记录文件所记载事件的详细程度。
-E 将信息送到标准错误输出设备。
-h 显示帮助。
-i<范围> 设置NetBIOS名称范围。
-I 指定服务器的IP地址。
-l<记录文件> 指定记录文件的名称。

-L 显示服务器端所分享出来的所有资源。
-M 可利用WinPopup协议，将信息送给选项中所指定的主机。
-n 指定用户端所要使用的NetBIOS名称。
-N 不用询问密码。
-O<连接槽选项> 设置用户端TCP连接槽的选项。
-p 指定服务器端TCP连接端口编号。
-R<名称解析顺序> 设置NetBIOS名称解析的顺序。
-s<目录> 指定smb.conf所在的目录。
-t<服务器字码> 设置用何种字符码来解析服务器端的文件名称。
-T 备份服务器端分享的全部文件，并打包成tar格式的文件。
-U<用户名称> 指定用户名称。
-W<工作群组> 指定工作群组名称。

----

shapecfg(shaper configuration)

功能说明：管制网络设备的流量。

语　　法：shapecfg attach [流量管制器][网络设备] 或 shapecfg speed [流量管制器][带宽]

补充说明：自Linux-2.15开始，便支持流量管制的功能。

参　　数：
attach 将流量管制器与实际的网络设备结合。
speed 设置流量管制器的对外传输带宽。

----

setserial

功能说明：设置或显示串口的相关信息。

语　　法：setserial [-abgGqvVz][设备][串口参数]

补充说明：setserial可用来设置串口或显示目前的设置。

参　　数：
-a 显示详细信息。
-b 显示摘要信息。
-g 显示串口的相关信息。
-G 以指令列表的格式来显示信息。
-q 执行时显示较少的信息。

-v 执行时显示较多的信息。
-V 显示版本信息。
-z 设置前，先将所有的标记归零。

----

samba

功能说明：Samba服务器控制。

语　　法：samba [start][stop][status][restart]

补充说明：samba为script文件，可启动，停止Samba服务器或回报目前的状态。

参　　数：
start 启动Samba服务器的服务。
stop 停止Samba服务器的服务。
status 显示Samba服务器目前的状态。
restart 重新启动Samba服务器。

----

pppstats(point to point protocol status)

功能说明：显示PPP连线状态。

语　　法：pppstats [-adrv][-c<执行次数>][-w<间隔秒数>][网络界面]

补充说明：利用pppstats指令可让你得知PPP连接网络的相关信息。

参　　数：
-a 显示绝对统计值。
-c<执行次数> 设置回报状况的次数。
-d 显示相对统计值。
-r 显示数据包压缩比率的统计值。
-v 显示VJTCP文件头的压缩效率统计值。
-w<间隔秒数> 设置显示统计信息的间隔时间。

----

pppsetup

功能说明：设置PPP连线。

语　　法：pppsetup

补充说明：这是Slackware发行版内附程序，它具有互动式的问答界面，让用户轻易完成PPP的连线设置。

----

ppp-off

功能说明：关闭ppp连线。

语　　法：ppp-of

补充说明：这是Slackware发行版内附的程序，让用户切断PPP的网络连线。

----

ping

功能说明：检测主机。

语　　法：ping [-dfnqrRv][-c<完成次数>][-i<间隔秒数>][-I<网络界面>][-l<前置载入>][-p<范本样式>][-s<数据包大小>][-t<存活数值>][主机名称或IP地址]

补充说明：执行ping指令会使用ICMP传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。

参　　数：
-d 使用Socket的SO_DEBUG功能。
-c<完成次数> 设置完成要求回应的次数。
-f 极限检测。
-i<间隔秒数> 指定收发信息的间隔时间。
-I<网络界面> 使用指定的网络界面送出数据包。
-l<前置载入> 设置在送出要求信息之前，先行发出的数据包。
-n 只输出数值。
-p<范本样式> 设置填满数据包的范本样式。
-q 不显示指令执行过程，开头和结尾的相关信息除外。
-r 忽略普通的Routing Table，直接将数据包送到远端主机上。
-R 记录路由过程。
-s<数据包大小> 设置数据包的大小。
-t<存活数值> 设置存活数值TTL的大小。
-v 详细显示指令的执行过程。

----

netstat

功能说明：显示网络状态。

语　　法：netstat [-acCeFghilMnNoprstuvVwx][-A<网络类型>][--ip]

补充说明：利用netstat指令可让你得知整个Linux系统的网络情况。

参　　数：
-a或--all 显示所有连线中的Socket。
-A<网络类型>或--<网络类型> 列出该网络类型连线中的相关地址。
-c或--continuous 持续列出网络状态。
-C或--cache 显示路由器配置的快取信息。
-e或--extend 显示网络其他相关信息。
-F或--fib 显示FIB。
-g或--groups 显示多重广播功能群组组员名单。
-h或--help 在线帮助。
-i或--interfaces 显示网络界面信息表单。
-l或--listening 显示监控中的服务器的Socket。
-M或--masquerade 显示伪装的网络连线。
-n或--numeric 直接使用IP地址，而不通过域名服务器。
-N或--netlink或--symbolic 显示网络硬件外围设备的符号连接名称。
-o或--timers 显示计时器。
-p或--programs 显示正在使用Socket的程序识别码和程序名称。
-r或--route 显示Routing Table。
-s或--statistice 显示网络工作信息统计表。
-t或--tcp 显示TCP传输协议的连线状况。
-u或--udp 显示UDP传输协议的连线状况。
-v或--verbose 显示指令执行过程。
-V或--version 显示版本信息。
-w或--raw 显示RAW传输协议的连线状况。
-x或--unix 此参数的效果和指定"-A unix"参数相同。
--ip或--inet 此参数的效果和指定"-A inet"参数相同。

----

mingetty

功能说明：精简版的getty。

语　　法：mingetty [--long-hostname][--noclear][tty]

补充说明：mingetty适用于本机上的登入程序。

参　　数：
--long-hostname 显示完整的主机名称。
--noclear 在询问登入的用户名称之前不要清楚屏幕画面。

----

mesg

功能说明：设置终端机的写入权限。

语　　法：mesg [ny]

补充说明：将mesg设置y时，其他用户可利用write指令将信息直接显示在您的屏幕上。

参　　数：
n 不允许气筒用户将信息直接显示在你的屏幕上。
y 允许气筒用户将信息直接显示在你的屏幕上。

----

ifconfig

功能说明：显示或设置网络设备。

语　　法：ifconfig [网络设备][down up -allmulti -arp -promisc][add〈地址〉][del〈地址〉][〈硬件地址〉][io_addr][irq][media〈网络媒介类型〉][mem_start〈内存地址〉][metric〈数目〉][mtu〈字节〉][netmask〈子网掩码〉][tunnel〈地址〉][-broadcast〈地址〉][-pointopoint〈地址〉][IP地址]

补充说明：ifconfig可设置网络设备的状态，或是显示目前的设置。

参　　数：
add〈地址〉 设置网络设备IPv6的IP地址。
del〈地址〉 删除网络设备IPv6的IP地址。
down 关闭指定的网络设备。
〈硬件地址〉 设置网络设备的类型与硬件地址。
io_addr 设置网络设备的I/O地址。
irq 设置网络设备的IRQ。
media〈网络媒介类型〉 设置网络设备的媒介类型。
mem_start〈内存地址〉 设置网络设备在主内存所占用的起始地址。
metric〈数目〉 指定在计算数据包的转送次数时，所要加上的数目。

mtu<字节> 设置网络设备的MTU。
netmask<子网掩码> 设置网络设备的子网掩码。
tunnel<地址> 建立IPv4与IPv6之间的隧道通信地址。
up 启动指定的网络设备。
-broadcast<地址> 将要送往指定地址的数据包当成广播数据包来处理。
-pointopoint<地址> 与指定地址的网络设备建立直接连线，此模式具有保密功能。
-promisc 关闭或启动指定网络设备的promiscuous模式。
[IP地址] 指定网络设备的IP地址。
[网络设备] 指定网络设备的名称。

----

httpd(http daemon)

功能说明：Apache HTTP服务器程序。

语　　法：httpd [-hlLStvVX][-c][-C][-d<服务器根目录>][-D<设定文件参数>][-f<设定文件>]

补充说明：httpd为Apache HTTP服务器程序。直接执行程序可启动服务器的服务。

参　　数：
-c 在读取配置文件前，先执行选项中的指令。
-C 在读取配置文件后，再执行选项中的指令。
-d<服务器根目录> 指定服务器的根目录。
-D<设定文件参数> 指定要传入配置文件的参数。
-f<设定文件> 指定配置文件。
-h 显示帮助。
-l 显示服务器编译时所包含的模块。
-L 显示httpd指令的说明。
-S 显示配置文件中的设定。
-t 测试配置文件的语法是否正确。
-v 显示版本信息。
-V 显示版本信息以及建立环境。
-X 以单一程序的方式来启动服务器。

----

getty(get teletypewriter)

功能说明：设置终端机模式，连线速率和管制线路。

语　　法：getty [-h][-d<组态配置文件>][-r<延迟秒数>][-t<超时秒数>][-w<等待字符串>][终端

机编号][连线速率<终端机类型><管制线路>] 或 getty [-c<定义配置文件>]

补充说明：getty指令是UNIX之类操作系统启动时所必须的3个步骤之一。

参　　数：
-c<定义配置文件> 指定定义配置文件，预设为/etc/gettydefs。
-d<组态配置文件> 指定组态配置文件，预设为/etc/conf.getty。
-h 当传输速率为0时就强制断线。
-r<延迟秒数> 设置延迟时间。
-t<超时秒数> 设置等待登入的时间。
-w<等待字符串> 设置等待回应的字符串。

----

efax

功能说明：收发传真。

语　　法：efax [-sw][-a][-c<调制解调器属性>][-d<驱动程序>][-f<字体文件>][-g<指令>][-h<传真标题字符串>][-i][-j][-k][-l<识别码>][-o<选项>][-q<错误次数>][-r<文件名>][-v<信息类型>][-x][-t<电话号码><传真文件>]

补充说明：支持Class 1与Class 2的调制解调器来收发传真。

参　　数：
-a 以指定的AT指令来接电话。
-c<调制解调器属性> 设置本机调制解调器的属性。
-d<驱动程序> 指定调制解调器驱动程序。
-f<字体文件> 使用指定的字体文件来建立传真标题。
-g<指令> 若接到的电话为数据，则执行指定的指令。
-h<传真标题字符串> 指定字符串为每页最前端的标题。
-i 在调制解调器进入传真模式前，传送AT指令到调制解调器。
-j 在调制解调器进入传真模式后，传送AT指令到调制解调器。
-k 在调制解调器离开传真模式前，传送AT指令到调制解调器。
-l<识别码> 设置本机调制解调器的识别码。
-o<选项> 使用非标准调制解调器时设置相关选项。
-q<错误次数> 接收传真时，当每页发生错误次数超过指定的数目时，要求对方重发。
-r<文件名> 在接收传真时，将每页分别保存成文件。
-v<信息类型> 选择要印出的信息类型。
-w 不要接听电话，等待OK或CONNECT的信号。
-x 使用UUCP格式的锁定文件来锁定调制解调器。
-t<电话号码><传真文件> 以<电话号码>中的号码来拨号，并将<传真文件>传真出去。

----

netconfig

功能说明：设置网络环境。

语　　法：netconfig

补充说明：这是Slackware发行版内附程序，它具有互动式的问答界面，让用户轻易完成网络环境的设置。

----

netconf

功能说明：设置各项网络功能。

语　　法：netconf

补充说明：netconf是Red Hat Linux发行版专门用来调整Linux各项设置的程序。

----

nc

功能说明：设置路由器。

语　　法：nc [-hlnruz][-g<网关...>][-G<指向器数目>][-i<延迟秒数>][-o<输出文件>][-p<通信端口>][-s<来源位址>][-v...][-w<超时秒数>][主机名称][通信端口...]

补充说明：执行本指令可设置路由器的相关参数。

参　　数：
-g<网关> 设置路由器跃程通信网关，最丢哦可设置8个。
-G<指向器数目> 设置来源路由指向器，其数值为4的倍数。
-h 在线帮助。
-i<延迟秒数> 设置时间间隔，以便传送信息及扫描通信端口。
-l 使用监听模式，管控传入的资料。
-n 直接使用IP地址，而不通过域名服务器。
-o<输出文件> 指定文件名称，把往来传输的数据以16进制字码倾倒成该文件保存。
-p<通信端口> 设置本地主机使用的通信端口。
-r 乱数指定本地与远端主机的通信端口。

-s<来源位址> 设置本地主机送出数据包的IP地址。
-u 使用UDP传输协议。
-v 显示指令执行过程。
-w<超时秒数> 设置等待连线的时间。
-z 使用0输入/输出模式，只在扫描通信端口时使用。

----

cu(call up)

功能说明：连接另一个系统主机。

语　　法：cu [dehnotv][-a<通信端口>][-c<电话号码>][-E<脱离字符>][-I<设置文件>][-l<外围设备代号>][-s<连线速率>][-x<排错模式>][-z<系统主机>][--help][-nostop][--parity=none][<系统主机>/<电话号码>]

补充说明：本指令可连接另一台主机，并采用类似拨号终端机的接口工作，也可执行简易的文件传输作业。

参　　数：
-a<通信端口>或-p<通信端口>或--port<通信端口> 使用指定的通信端口进行连线。
-c<电话号码>或--phone<电话号码> 拨打该电话号码。
-d 进入排错模式。
-e或--parity=even 使用双同位检查。
-E<脱离字符>或--escape<脱离字符> 设置脱离字符。
-h或--halfduple 使用半双工模式。
-I<配置文件>或--config<配置文件> 指定要使用的配置文件。
-l<外围设备代号>或--line<外围设备代号> 指定某项外围设备，作为连接的设备。
-n或--prompt 拨号时等待用户输入电话号码。
-o或--parity=odd 使用单同位检查。
-s<连线速率>或--speed<连线速率>或--baud<连线速率>或-<连线速率> 设置连线的速率，单位以鲍率计算。
-t或--maper 把CR字符置换成LF+CR字符。
-v或--version 显示版本信息。
-x<排错模式>或--debug<排错模式> 使用排错模式。
-z<系统主机>或--system<系统主机> 连接该系统主机。
--help 在线帮助。
--nostop 关闭Xon/Xoff软件流量控制。
--parity=none 不使用同位检查。

----

arpwatch(ARP watcher)

功能说明：监听网络上ARP的记录。

语　　法：arpwatch [-d][-f<记录文件>][-i<接口>][-r<记录文件>]

补充说明：ARP(Address Resolution Protocol)是用来解析IP与网络装置硬件地址的协议。arpwatch可监听区域网络中的ARP数据包并记录，同时将监听到的变化通过E-mail来报告。

参　　数：
-d 启动排错模式。
-f<记录文件> 设置存储ARP记录的文件，预设为/var/arpwatch/arp.dat。
-i<接口> 指定监听ARP的接口，预设的接口为eth0。
-r<记录文件> 从指定的文件中读取ARP记录，而不是从网络上监听。

----

apachectl(Apache control interface)

功能说明：可用来控制Apache HTTP服务器的程序。

语　　法：apachectl [configtest][fullstatus][graceful][help][restart][start][status][stop]

补充说明：apachectl是slackware内附Apache HTTP服务器的script文件，可供管理员控制服务器，但在其他Linux的Apache HTTP服务器不一定有这个文件。

参　　数：
configtest 检查设置文件中的语法是否正确。
fullstatus 显示服务器完整的状态信息。
graceful 重新启动Apache服务器，但不会中断原有的连接。
help 显示帮助信息。
restart 重新启动Apache服务器。
start 启动Apache服务器。
status 显示服务器摘要的状态信息。
stop 停止Apache服务器。

----

minicom

功能说明：调制解调器通信程序。

语　　法：minicom [-8lmMostz][-a][-c][-C<取文件>][-d<编号>][-p<模拟终端机>][-S][配置文件]

补充说明：minicom是一个相当受欢迎的PPP拨号连线程序。

参　　数：
-8 不要修改任何8位编码的字符。
-a 设置终端机属性。
-c 设置彩色模式。
-C<取文件> 指定取文件，并在启动时开启取功能。
-d<编号> 启动或直接拨号。
-l 不会将所有的字符都转成ASCII码。
-m 以Alt或Meta键作为指令键。
-M 与-m参数类似。
-o 不要初始化调制解调器。
-p <模拟终端机> 使用模拟终端机。
-s 开启程序设置画面。
-S 在启动时，执行指定的script文件。
-t 设置终端机的类型。
-z 在终端机上显示状态列。
[配置文件] 指定minicom配置文件。

----

dip

功能说明：IP拨号连接。

语　　法：dip [-aikltv][-m][-p<协议>][拨号script文件]

补充说明：dip可控制调制解调器，以拨号IP的方式建立对外的双向连接。

参　　数：
-a 询问用户名称与密码。
-i 启动拨号服务器功能。
-k 删除执行中的dip程序。
-l 指定要删除的连线，必须配合-k参数一起使用。
-m 设置最大传输单位，预设值为296。
-p<协议> 设置通信协议。
-t 进入dip的指令模式。
-v 执行时显示详细的信息。

----

dnsconf(dns configurator)

功能说明：设置DNS服务器组态。

语　　法：dnsconf [--deldomain<域>][--delsecondary<域>][--newdomain<域>][--set<主机>][--setcname<主机>][--setmx<域><主机>][--setns<域><主机>][--unset<主机>]

补充说明：dnsconf实际上为linuxconf的符号连接，提供图形截面的操作方式，供管理员管理DNS服务器。

参　　数：
--deldomain<域> 删除域。
--delsecondary<域> 删除次级域。
--newdomain<域> 新增域。
--set<主机> 新增主机记录。
--setcname<主机> 设置。
--setmx<域><主机> 指定域的邮件主机。
--setns<域><主机> 指定域的DNS服务器。
--unset<主机> 删除DNS中某台主机的记录。

<hr>

linux下各种分区的自动挂载和卸载

目前实现分区自动挂载/卸载的方法(我知道的)有两种,一种是supermount,一种是automount,supermount要重新编译内核,而且不稳定,这里就不介绍了.
我介绍一下automount的使用方法:
首先确认你的linux已经安装autofs这个包.
如果你不想每次开机都手动启动autofs服务,就运行setup,在System services里把
autofs选上.

编辑/etc/auto.master,把# /misc /etc/auto.misc --timeout=60
这一行的#号去掉.

编辑/etc/auto.misc
照下面的样子加入要自动mount的分区的配置.
(不要全跟我一样,我只是举个例子)

名字 分区类型 设备名
cdrom -fstype=auto : /dev/cdrom
floppy -fstype=auto :/dev/fd0
c -fstype=auto,iocharset=cp936 :/dev/hda1
d -fstype=auto,iocharset=cp936 :/dev/hda5
hda7 -fstype=auto :/dev/hda7
.
.
.
iocharset=cp936是针对windows分区的，linux分区不要加这个
修改完毕后,重新启动一下autofs服务,方法/etc/init.d/autofs restart
现在你就可以很方便的访问任何一个分区,包括光驱,软驱了.
比如你在终端里输入cd /misc/c,那么系统就会自动把/dev/hda1分区mount
到/misc/c目录上.cd /misc/cdrom就可以访问光驱.如果60妙内你没再访问自动
mount上的分区,那么系统自动会把它umount掉.

/misc目录下的cdrom,floppy,c....等目录是系统自动建的,umount后这些目录会被
自动删除.不要手动在/misc目录下建任何目录或文件.

为了避免于fstab里的自动mount分区冲突,把/etc/fstab里的
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
这两行删除掉.

现在你可以在桌面上建一些快捷键,让它链接到/misc/xxx,你就可以很方便地
访问各个分区了.

Linux下其它格式文件系统的自动挂装

不同的操作系统使用不同的文件系统格式，MS-DOS支持FAT16文件系统，Windows98支持FAT16、FAT32文件系统，WindowsNT支持FAT16、NTFS文件系统，Windows2000则支持FAT16、FAT32、NTFS三种文件系统格式，而Linux差不多支持所有的文件系统格式，但一般使用ext2文件系统。

对于普通的PC用户，使用的大多是Microsoft的windows98操作系统，如果想同时使用Linux操作系统的话，一般使用多重启动。这时，用户可能希望在Linux下访问Windows文件，比如原来Windows下的mp3、电影等等。通常，有多种方法实现Linux下对其它文件系统的访问，下面就以在Linux下对Windows98文件系统的访问为例进行介绍：

（一）在安装Linux时进行设置。由于Linux和Windows98两种操作系统使用不同的文件格式，所以在Windows98下安装Linux必须为Linux建立单独的分区，安装过程中有设置分区的步骤，此时，不但可以建立Linux分区，还可以对原FAT文件系统进行挂装，因为Linux把磁盘设备也看成是特殊的文件，这时如果为Windows98操作系统的逻辑分区C、D分别设立如/DOSC、/DOSD的挂装点，那么在安装完Linux后就可以

在/DOSC、/DOSD目录下访问原来C、D分区上的文件了。

（二）如果在安装时没有为Windows98分区设立挂装点，也可以在安装完成后进入Linux运行mount命令（必须以root身份登录）挂装，mount命令的具体格式是：

mount〔-afFhnrvVm〕〔-l<标签>〕〔-o<选项>〕〔-t<文件系统>〕〔设备名称〕〔挂入点〕

比如，用户的Windows98在系统IDE设备上的第一个分区上，如果要将它挂装在/DOSC下，可以使用下面的命令：

mount -t vfat /dev/hda1 /dosc

使用mount挂装设备时会记录信息在/etc/mtab文件中，运行unmount命令时将记录清除。

（三）手动修改/etc/fstab文件。/etc/fstab文件的内容被用来在Linux 下自动挂装各种文件系统，文件中的每一行都提供了一种设备的信息，这种设备可以被挂装在Linux文件系统下的一个目录中。在Linux启动过程中，init进程执行一个脚本文件，该脚本调用带有-a参数的mount命令，用mount来读/etc/fstab，并挂装所有列出的文件系统（带noauto选项的除外）。下面是一个典型的/etc/fstab文件(其中Windows98安装在第一块硬盘上，Linux安装在第二块硬盘上)：

```
ABEL=/ / ext2 defaults 1 1
/dev/hda1 /dosc vfat defaults 0 0
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
none /proc proc defaults 0 0
none /devpts devpts sid=5,mode=620 0 0
/dev/hdb5 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kud2ure 0/0
```

/etc/fstab文件的第一列是设备名，第二列是挂装点，第三列表示在设备上的文件系统的类型，第四列是应用于特定设备的一组选项，通常为defaults，表示的含义有：这个设备在引导阶段被挂装、只有root用户可以挂装它、挂装后可以进行读或写操作，此选项如是noauto，则表示引导时该设备不会被自动挂装，而user选项表示任何用户都可以挂装该设备。以上面的etc/fstab文件为例，如果想在Linux下挂装Windows98操作系统的D分区(假设挂装点为/dosd)，则可以在上文件中加入一行：

```
/dev/hda5 /dosd vfat defaults 0 0
```

然后存盘，重新启动后就可以访问Windows98操作系统的D分区了。

补充：fstab 栏位说明

第一栏(fs_spec)： 实际的 device 名称

第二栏(fs_file)： 对应到的目录结构(mount point)

第三栏(fs_vfstype)：该 partition 的档案系统，常见的有：

minix、ext、ext2、msdos、iso9660、nfs、swap

第四栏(fs_mntops)： 在 mount 时的参数

第五栏(fs_freq)： 在使用 dump 时是否记录，不需要则输入0

第六栏(fs_passno)： 决定在开机时执行 fsck 的先后顺序
<hr>

上一页 目录 下一页

Linux下软件安装方法

对于Linux初学者来说，安装一个很小的软件恐怕都是一件很让人头疼的事，因为在Linux下安装软件不像在 Windows中那样简单。在Linux中大多数软件提供的是源代码，而不是现成的可执行文件，这就要求用户根据自己系统的实际情况和自身的需要来配置、编译源程序后，软件才能使用。多数初学者往往不知道该如何进行配置和编译就盲目地运行一些有执行属性的文件或者机械地运行“make”、“make install”之类的命令。结果呢？是软件没装成，自己急出一身汗，后果严重的还会破坏系统的稳定性。下面笔者将安装软件方面的一些规律写出来与大家分享。

目前流行的软件包有两种比较常见的形式，一种是RPM包的形式，另一种是压缩成*.tar.gz的形式。本文将讨论这两种形式的软件包在文本环境和图形环境（X Window）下不同的安装方法。

文本环境下的软件安装

一、安装简便的RPM包

RPM是RedHat Package Manager的缩写，它只能使用在安装了RPM软件的系统中，RedHat Linux和Turbo Linux中已经使用了它。这种结构的包使用起来还是很方便的，只要记住几条简单的命令和参数就可以方便地使用：

#〉rpm [options] filename.rpm

其中常用的options包括：

-i： 安装软件

-e： 卸载软件

-q：查看软件安装的信息和状态

-U： 升级现有软件

例如安装软件时，可以使用如下命令：“rpm -i filename.rpm”，软件安装在什么地方、是怎么安装的都不需要用户操心，RPM可以帮助用户管理。由于RPM使用方便，很多软件都有RPM版本。如果想使用RPM形式的软件就要首先下载一个RPM管理软件。在ftp://ftp.rpm.org/pub/rpm/dist/rpm-version

二、安装需要编译的.tar.gz包

由于RPM包使用的局限性（必须安装RPM），目前更多的软件使用的是源代码形式的.tar.gz包。这种软件的安装通常要经过解压缩、软件配置、软件编译及安装的过程。

解压缩通常有两种命令方法：一种是“gunzip filename-VERSION-OS.tar.gz | tar xvf -”，它实质是两条命令“gunzip filename-VERSION-OS.tar.gz”和“tar xvf filename-VERSION-OS.tar”；另一种是“tar xzvf filename-VERSION-OS.tar.gz”。 一般来说在软件解压缩后会生成一个目录filename-VERSION-OS。

软件的配置、编译、安装是最让初学者望而生畏的事了，但笔者认为掌握一些规律还是不难的。一般来说，在解压缩生成的目录中都会有名为Readme、Rnstall或Readme.install之类的文件。这些文件通常会对软件的功能、特性、版权许可、安装以及相关知识加以介绍，并且会提到关于安装的方法和步骤。举

例来说：在apache_1.3.6的install文件中说明了如下内容（此处只列出总的条目，具体内容省略）：

Installing the Apache 1.3 HTTP server with APACI

==============================

1.Overview for the impatient（概括说明配置的全过程）

$./configure--prefix=PREFIX

$make

$make install

$PREFIX/bin/apachectl start

2.Requirements（需要的条件）

3.Configuring the source tree(配置的参数说明)

4.Building the package（编译软件的方法）

5.Installing the package（安装软件的方法）

6.Testing the package（软件测试）

理解并能熟练使用这些说明文件后，就可以利用一些规律来安装大多数的软件。对于那些没有说明文件的软件（当然这种情况比较少见），这些规律通常也是适用的。一般来说，与安装软件有直接关系的文件只有两个：configure 、Makefile。

其中，configure文件具有可执行的属性，是用来配置软件的，它的参数比较多，用法也比较灵活。当然，不同的软件参数也不相同，这时候就需要借助它的help参数，运行下面的命令就会让你感到豁然开朗：

#〉 ./configure -help

Usage: configure [options]

Options: [defaults in brackets after descriptions]

General options:

--quiet, --silent do not print messages

--verbose,-v print even more messages

--sha [=DIR] switch to a shadow tree (under DIR) for building

Stand-alone options:

--help,-h print this message

--show-layout print installation path layout (check and debug)

Installation layout options:

--with-layout=[F:]ID use installation path layout ID (from file F)

--target=TARGET install name-associated files using basename TARGET

……

接下来，就可以运行“./configure [options]”来配置该软件。注意，命令行中的“./”非常重要，它告诉系统要运行的命令就在当前目录下（否则系统就会到$path变量指定的路径下去执行命令）。执行命令后可以生成Makefile文件或者修改已有的文件配置。

Makefile文件通常是用来编译和安装软件的。运行make命令时系统会自动根据Makefile文件中的设置对软件进行编译和安装。make命令有时还可以带一些参数，如：all、build、config、install等。具体要带哪个参数可以参看Makefile文件。在Linux中绝大部分文件是文本文件，Makefile就是一个shell程序（Linux中shell程序与DOS中的批处理文件有很多相似之处，当然功能要强得多），很容易读懂，尤其是编译时可带的参数都会明确写出，例如：

```
##========================

## Targets

##========================

# default target

all: build

##------------------------

## Build Target

##------------------------

# build the package

build:

……
```

```
# the non-verbose variant for package maintainers

build-quiet:

@$(MAKE) -f $(TOP)/$(MKF) $(MFLAGS) $(MFWD) QUIET=1 build

# build the additional support stuff

build-support:

……

##------------------------

## Installation Targets

## ------------------------

# the install target for installing the complete Apache

# package. This is implemented by running subtargets for the

# separate parts of the installation process.

install:

……

# the non-verbose variant for package maintainers

install-quiet:

@$(MAKE) -f $(TOP)/$(MKF) $(MFLAGS) $(MFWD) QUIET=1 install

# create the installation tree

install-mktree:

……
```

上面这段代码是apache_1.3.6的Makefile文件的一部分，从这段程序可以看出all参数表示完全编译（缺省参数）。此外，编译时还可以带build、build-quiet、build-surpport等参数；安装时可以带install、install-quiet、install-surpport等参数。它们的功能分别在“#”表示的注释中进行了说明。需要额外说明的是，有些软件（例如Linux的内核升级程序）不用configure命令来配置软件，而是用make config来完成这项工作，所以，具体使用哪种方法要具体问题具体分析。

图形界面下的软件安装

在图形环境下，同样可以弹出一个仿真终端以文本的方法来安装软件，但那样就太笨了，因为在X Window中有一些简单的方法可以帮您完成软件安装。下面笔者以RedHat 6.0为例做介绍。

一、图形界面下安装.tar.gz包

在X Window下，安装这种形式的包简化程度并不大，只是在解压缩软件时方便一些。用鼠标左键双击要安装的软件包，系统就会自动生成一个目录——filename.tar.gz#utar，在这个目录下就有你要解压缩的内容——filename目录。将该目录拷贝到你要解压缩的目录下，解压缩的工作就这样简单地完成了。不过，剩下的工作还要弹出一个仿真终端以文本的方法来完成。

二、图形界面下安装RPM包

在X Window中你要安装、升级、卸载和查询一个RPM软件包实在是太容易了。以Redhat 6.0为例，它的X Window中有一个Gnome RPM软件可以完成上面提到的一切工作。

点击“Start”→“System”→“GnomeRPM”，可以运行该软件。它将安装好的RPM包形式的软件按照功能分在Amusements、Application、Development、Document、Extension、Extentions、Networking、System Environment、UserInterface、X11等几个树形目录中，每个目录中有相应的文件图标和名称。要安装或升级软件，只要点击工具栏的相应按钮就会弹出打开文件的对话框，选中你要安装的RPM文件，单击“OK”按钮，一切大功告成；卸载软件时，需要根据分类找到该软件的图标，点击右键，选Uninstall就可以了。如果你会在Win 95/98中查找文件的话，在Linux中查找已经安装的软件也就不难了。这个软件可以到下面的站点下载：ftp.gnome.org/pub/GNOME/stable/

sources/gnorpm。

　几点注意事项

一、安装方法的适用范围

上面提到的软件安装方法并不是在任何版本的Linux上都适用。笔者认为，.tar.gz包的安装方法适用于各种版本的Linux，而RPM包则有一定的局限性。

目前常见的各种Linux发行版本中，如：Redhat 6.0、Turbo Linux 3.0.2、Xteam 3.0等都支持RPM包。如果你想知道你使用的Linux是否支持RPM包，只要运行一下“rpm --help”命令就知道了。不过，对于不支持RPM包的版本，可以安装一个RPM管理软件。

二、容易出现的问题

在安装软件时，一定要保证你对用到的软件包有访问权限。当然如果你是root就没问题了。但如果你真的是root你就需要注意另一个问题，由于root的权限过高，所以在安装软件时，要防止对系统其它软件造成误操作（在使用rm 等“危险”命令时，尤其要注意）。

另一个容易出现的问题是，在卸载RPM包的软件时要慎重，因为很多软件之间是相互关联的，你卸载的软件很可能是其它软件要用到的，要防止由于卸载了一个软件而影响另一个软件的正常使用。所以笔者建议，初学者对于与系统运行有关的软件尽量不要删除（对于游戏、应用软件一类的软件不必有太多顾

虑）。等你成为一名经验丰富的系统管理员时，你就能灵活地处理这些问题了。
<hr>

上一页 目录 下一页

上一页 目录 下一页

Linux怎么安装东西

1、源代码包的安装

gzip -d apache_1.3.20.tar.gz （解压）

tar xvf apache_1.3.20.tar （解包）

cd apache_1.3.20

./configure （配置） ----./configure --help（查看configure选项）

make （编译）

make install （安装）

make clean （卸载）

注：典型的源代码包可以这样安装，但不都是这样，如webmin

要执行其目录下./setup.sh进入交互式配置安装

卸载用uninstall程序

具体如果不清楚看要安装的包下的README文件

2、RPM包的安装

RPM软件包的一个例子：

foo-1.0-1.i386.rpm

其中包括软件包的名称(foo)，版本号(1.0)，发行号(1)，和硬件平台(i386)。

（1）安装

rpm -ivh foo-1.0-1.i386.rpm

foo

#######################

A. 软件包已被安装

rpm -ivh foo-1.0-1.i386.rpm

foo package foo-1.0-1 is already installed

error: foo-1.0-1.i386.rpm cannot be installed

如果你仍旧要安装该软件包，你可以在命令行上使用--replacepkgs 选项，这将忽略该错误信息。

B. 文件冲突

如果要安装的软件包中有一个文件已在安装其它软件包时安装，会出现以下错误信息：

rpm -ivh foo-1.0-1.i386.rpm

foo /usr/bin/foo conflicts with file from bar-1.0-1

error: foo-1.0-1.i386.rpm cannot be installed

要想让RPM 忽略该错误信息, 请使用--replacefiles 命令行选项

C.未解决依赖关系

RPM软件包可能依赖于其它软件包, 在安装了特定的软件包之后才能安装该软件包。

rpm -ivh bar-1.0-1.i386.rpm
failed dependencies:
foo is needed by bar-1.0-1
你必须安装完所依赖的软件包，才能解决这个问题。如果想强制安装（但是，这样安装后的软件包未必能正常运行)，请使用-nodeps 命令行选项。

（2） 卸载
rpm -e foo
注意：这里使用软件包的名字foo，而不是软件包文件的名字“foo-1.0-1.i386.rpm”.
如果其它软件包依赖于你要卸载的软件包，卸载时则会产生错误信息。如：
rpm -e foo
removing these packages would break dependencies:
foo is needed by bar-1.0-1
要想RPM忽略该错误信息继续卸载的话（但是，依赖于该软件包的程序可能无法运行)，请使用-nodeps 命令行选项。
（3）升级
rpm -Uvh foo-2.0-1.i386.rpm
foo
###############################
当使用旧版本的软件包来升级新版本的软件时，会产生以下错误信息：
rpm -Uvh foo-1.0-1.i386.rpm
foo package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
要使RPM 坚持这样“升级”，可使用--oldpackage 命令行参数。
（4）查询
rpm -q foo
foo-2.0-1
软件包指定选项：
-a 查询所有已安装的软件包.
-f <file> 将查询包含有文件<file>的软件包
-p <packagefile>
查询软件包文件名为<packagefile>的软件包
信息选择选项：
-i 显示软件包信息，如描述，发行号，尺寸，构建日期，安装日期，平台，以及其它各类信息。
-l 显示软件包中的文件列表。
-s 显示软件包中所有文件的状态。
-d 显示被标注为文档的文件列表(man 手册，info 手册，README's，etc).
-c 显示被标注为配置文件的文件列表。这些是要在安装完毕以后加以定制的文件(sendmail.cf，passwd，inittab，etc)。
对于那些要显示文件列表的文件,可以增加-v 命令行选项以获得如同 ls -l 格式的输出。
（5）验证
验证软件包是通过比较软件包中安装的文件和软件包中的原始文件信息来进行的。除了其它一些东西，验证主要是比较文件的尺寸，MD5 校验码，文件权限，类型，属主和用户组等。
rpm -V命令用来验证一个软件包，如
rpm -V foo
验证包含特定文件的软件包:
rpm -Vf /bin/vi

验证所有已安装的软件包:
rpm -Va
根据一个RPM来验证某个软件包:
rpm -Vp foo-1.0-1.i386.rpm
如果你担心RPM数据库已被破坏，就可以使用这种方式。如果一切校验均正常将不会产生任何输出。如果有不一致的地方，就会显示出来。
输出格式是8位长字符串，c 用以指配置文件，接着是文件名. 8位字符的每一个用以表示文件与RPM数据库中一种属性的比较结果。“.”（点)表示测试通过。
以下字符表示某种测试的失败：
5 MD5 校验码
S 文件尺寸
L 符号连接
T 文件修改日期
D 设备
U 用户
G 用户组
M 模式e (包括权限和文件类型)
如果有信息输出，应当认真加以考虑，是删除，重新安装，还是修正出现的问题。
（6）RPM应用的几个例子
A. 如你误删了一些文件, 但是不能肯定到底删除了哪些文件。如果你想验证一下整个系统看看都丢失了哪些文件的话，可以键入：
rpm －Va
B.若是一些文件丢失了或已被损坏, 就可以重新安装或先卸载再安装该软件包。如果碰到了一个自己不认识的文件，要想查处它属于哪个软件包，可以输入以下命令：
rpm -qf /usr/X11R6/bin/xjewel
而输出的结果会是：
xjewel-1.6-1
C. 如果发生综合以上两个例子的情况，如文/usr/bin/paste出了问题。你想验证一下拥有该文件的软件包，可又不知道软件包的名字，这时可以简单的键入：
rpm -Vf /usr/bin/paste
这样相应的软件包就会被验证。
D. 如果你想了解一个正在使用的程序的详细信息，可以键入如下命令来获得拥有该程序的软件包中的文档信息：
rpm -qdf /usr/bin/ispell
输出结果为：
/usr/man/man4/ispell.4
/usr/man/man4/english.4
…………………………………………………………

E. 如果你发现了一个新的koules RPM，但是不知道它是什么东西，可以键入如下命令：
rpm -qip koules-1.2-2.i386.rpm
F. 现在你想了解koules RPM 所安装的文件。可以键入：
rpm -qlp koules-1.2-2.i386.rpm
图形管理工具：kpackage

3、shell或java脚本安装
基于图形界面的安装，一般基于SHELL或Java语言编写，主要应用于一些办公软件和制图软件及安装程

序，如staroffice、oracle的安装，很容易，和windows一样
<hr>

上一页 目录 下一页

linux指令

cat cd
chmod chown
cp cut
名称：cat
使用权限：所有使用者
使用方式：cat [-AbeEnstTuv] [--help] [--version] fileName
说明：把档案串连接后传到基本输出（萤幕或加 > fileName 到另一个档案）
参数：
-n 或 --number 由 1 开始对所有输出的行数编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行
-v 或 --show-nonprinting
范例：
cat -n textfile1 > textfile2 把 textfile1 的档案内容加上行号后输入 textfile2 这个档案里
cat -b textfile1 textfile2 >> textfile3 把 textfile1 和 textfile2 的档案内容加上行号（空白行不加）之后将内容附加到 textfile3

名称 : cd
使用权限 : 所有使用者
使用方式 : cd [dirName]
说明 : 变换工作目录至 dirName。 其中 dirName 表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的 home directory (也就是刚 login 时所在的目录)。
另外，"~" 也表示为 home directory 的意思，"." 则是表示目前所在的目录，".." 则表示目前目录位置的上一层目录。
范例 : 跳到 /usr/bin/ :
cd /usr/bin
跳到自己的 home directory :
cd ~
跳到目前目录的上上两层 :
cd ../..

cd ../..

指令名称 : chmod
使用权限 : 所有使用者
使用方式 : chmod [-cfvR] [--help] [--version] mode file...
说明 : Linux/Unix 的档案存取权限分为三级 : 档案拥有者、群组、其他。利用 chmod 可以藉以控制档案如何被他人所存取。
把计 :
mode : 权限设定字串，格式如下 : [ugoa...][[+-=][rwxX]...][,...]，其中u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。
-c : 若该档案权限确实已经更改，才显示其更改动作
-f : 若该档案权限无法被更改也不要显示错误讯息
-v : 显示权限变更的详细资料
-R : 对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)
--help : 显示辅助说明
--version : 显示版本
范例 :将档案 file1.txt 设为所有人皆可读取 :
chmod ugo+r file1.txt
将档案 file1.txt 设为所有人皆可读取 :
将档案 file1.txt 设为所有人皆可读取 :
chmod a+r file1.txt
将档案 file1.txt 与 file2.txt 设为该档案拥有者，与其所属同一个群体者可写入，但其他以外的人则不可写入 :
chmod ug+w,o-w file1.txt file2.txt
将 ex1.py 设定为只有该档案拥有者可以执行 :
chmod u+x ex1.py
将目前目录下的所有档案与子目录皆设为任何人可读取 :
chmod -R a+r *
此外chmod也可以用数字来表示权限如 chmod 777 file
语法为：chmod abc file
其中a,b,c各为一个数字，分别表示User、Group、及Other的权限。
r=4，w=2，x=1
若要rwx属性则4+2+1=7；
若要rw-属性则4+2=6；
若要r-x属性则4+1=7。
范例：
chmod a=rwx file
和
chmod 777 file
效果相同
chmod ug=rwx,o=x file
和

和
chmod 771 file
效果相同
若用chmod 4755 filename可使此程式具有root的权限

指令名称 : chown
使用权限 : root
使用方式 : chmod [-cfhvR] [--help] [--version] user[:group] file...
说明 : Linux/Unix 是多人多工作业系统，所有的档案皆有拥有者。利用 chown 可以将档案的拥有者加以改变。一般来说，这个指令只有是由系统管理者(root)所使用，一般使用者没有权限可以改变别人的档案拥有者，也没有权限可以自己的档案拥有者改设为别人。只有系统管理者(root)才有这样的权限。
把计 :
user : 新的档案拥有者的使用者 IDgroup : 新的档案拥有者的使用者群体(group)-c : 若该档案拥有者确实已经更改，才显示其更改动作-f : 若该档案拥有者无法被更改也不要显示错误讯息-h : 只对于连结(link)进行变更，而非该 link 真正指向的档案-v : 显示拥有者变更的详细资料-R : 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)--help : 显示辅助说明--version : 显示版本
范例 :
将档案 file1.txt 的拥有者设为 users 群体的使用者 jessie :
chown jessie:users file1.txt
将目前目录下的所有档案与子目录的拥有者皆设为 users 群体的使用者
lamport :
lamport :
chown -R lamport:users *

名称：cp
使用权限：所有使用者
使用方式：
cp [options] source dest
cp [options] source... directory
说明：将一个档案拷贝至另一档案，或将数个档案拷贝至另一目录。
把计
-a 尽可能将档案状态、权限等资料都照原状予以复制。
-r 若 source 中含有目录名，则将目录下之档案亦皆依序拷贝至目的地。
-f 若目的地已经有相同档名的档案存在，则在复制前先予以删除再行复制。
范例：
将档案 aaa 复制(已存在)，并命名为 bbb :
cp aaa bbb
将所有的C语言程式拷贝至 Finished 子目录中 :
cp *.c Finished

名称：cut
使用权限：所有使用者
用法：cut -cnum1-num2 filename
说明：显示每行从开头算起 num1 到 num2 的文字。

范例：

shell>> cat example

shell>> cat example

test2

this is test1

shell>> cut -c0-6 example ## print 开头算起前 6 个字元

test2

this i

名称 : find

用法 : find

使用说明 :

将档案系统内符合 expression 的档案列出来。你可以指要档案的名称、类别、时间、大小、权限等不同资讯的组合，只有完全相符的才会被列出来。

find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ! 之前的部份为 path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用 -print 为预设 expression

expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。

-mount, -xdev : 只检查和指定目录在同一个档案系统下的档案，避免列出其它档案系统中的档案

-amin n : 在过去 n 分钟内被读取过

-anewer file : 比档案 file 更晚被读取过的档案

-atime n : 在过去 n 天过读取过的档案

-cmin n : 在过去 n 分钟内被修改过

-cnewer file :比档案 file 更新的档案

-ctime n : 在过去 n 天过修改过的档案

-ctime n : 在过去 n 天过修改过的档案

-empty : 空的档案-gid n or -group name : gid 是 n 或是 group 名称是 name

-ipath p, -path p : 路径名称符合 p 的档案，ipath 会忽略大小写

-name name, -iname name : 档案名称符合 name 的档案。iname 会忽略大小写

-size n : 档案大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w 是二个位元组。-type c : 档案类型是 c 的档案。

d: 目录

c: 字型装置档案

b: 区块装置档案

p: 具名贮列

f: 一般档案

l: 符号连结

s: socket

-pid n : process id 是 n 的档案

你可以使用 ( ) 将运算式分隔，并使用下列运算。

exp1 -and exp2

! expr

-not expr

exp1 -or exp2

exp1, exp2

范例:
将目前目录及其子目录下所有延伸档名是 c 的档案列出来。
将目前目录及其子目录下所有延伸档名是 c 的档案列出来。
find . -name "*.c"
将目前目录其其下子目录中所有一般档案列出
find . -ftype f
将目前目录及其子目录下所有最近 20 分钟内更新过的档案列出
find . -ctime -20

名称：less
使用权限：所有使用者
使用方式：
less [Option] filename
说明：
less 的作用与 more 十分相似，都可以用来浏览文字档案的内容，不同的是 less 允许使用者往回卷动
以浏览已经看过的部份，同时因为 less 并未在一开始就读入整个档案，因此在遇上大型档案的开启时，会比一般的文书编辑器(如 vi)来的快速。
范例：

指令名称 : ln
使用权限 : 所有使用者
使用方式 : ln [options] source dist，其中 option 的格式为 :
[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]
[--help] [--version] [--]
说明 : Linux/Unix 档案系统中，有所谓的连结(link)，我们可以将其视为档案的别名，而连结又可分为两种 : 硬连结(hard link)与软连结(symbolic link)，
的别名，而连结又可分为两种 : 硬连结(hard link)与软连结(symbolic link)，
硬连结的意思是一个档案可以有多个名称，而软连结的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬连结是存在同一个档案系统中，而软连结却可以跨越不同的档案系统。
ln source dist 是产生一个连结(dist)到 source，至于使用硬连结或软链结则由参数决定。
不论是硬连结或软链结都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。
-f : 链结时先将与 dist 同档名的档案删除-d : 允许系统管理者硬链结自己的目录-i : 在删除与 dist 同档名的档案时先进行询问-n : 在进行软连结时，将 dist 视为一般的档案-s : 进行软链结(symbolic link)-v : 在连结之前显示其档名-b : 将在链结时会被覆写或删除的档案进行备份-S SUFFIX : 将备份的档案都加上 SUFFIX 的字尾-V METHOD : 指定备份的方式--help : 显示辅助说明--version : 显示版本
范例 :
将档案 yy 产生一个 symbolic link : zz
ln -s yy zz
将档案 yy 产生一个 hard link : zz
ln yy xx

名称：locate
使用权限：所有使用者
使用方式： locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-e] [-f] <[-l] [-c]
<[-U] [-u]>
locate [-Vh] [--version] [--help]
说明：
locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的资料库，之后当寻找时就只需查询这个资料库，而不必实际深入档案系统之中了。
在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用
locate your_file_name
的型式就可以了。 参数：
-u
-U
建立资料库，-u 会由根目录开始，-U 则可以指定开始的位置。
-e
将
排除在寻找的范围之外。
-l
如果 是 1．则启动安全模式。在安全模式下，使用者不会看到权限无法看到的档案。这会始速度减慢，因为 locate 必须至实际的档案系统中取得档案的权限资料。
料。
-f
将特定的档案系统排除在外，例如我们没有到理要把 proc 档案系统中的档案放在资料库中。
-q
安静模式，不会显示任何错误讯息。
-n
至多显示 个输出。
-r
使用正规运算式 做寻找的条件。
-o
指定资料库存的名称。
-d
指定资料库的路径
-h
显示辅助讯息
-v
显示更多的讯息
-V
显示程式的版本讯息 范例：

locate chdrv : 寻找所有叫 chdrv 的档案
locate -n 100 a.out : 寻找所有叫 a.out 的档案，但最多只显示 100 个
locate -u : 建立资料库
locate -u : 建立资料库

名称 : ls
使用权限 : 所有使用者
使用方式 : ls [-alrtAFR] [name...]
说明 : 显示指定工作目录下之内容（列出目前工作目录所含之档案及子目录)。
-a 显示所有档案及目录 (ls内定将档案名或目录名称开头为"."的视为隐藏档，不会列出)
-l 除档案名称外，亦将档案型态、权限、拥有者、档案大小等资讯详细列出
-r 将档案以相反次序显示(原定依英文字母次序)
-t 将档案依建立时间之先后次序列出
-A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)
-F 在列出的档案名称后加一符号；例如可执行档则加 "*", 目录则加 "/"
-R 若目录下有档案，则以下之档案亦皆依序列出
范例:
列出目前工作目录下所有名称是 s 开头的档案，愈新的排愈后面 :
ls -ltr s*
将 /bin 目录以下所有目录及档案详细资料列出 :
ls -lR /bin
列出目前工作目录下所有档案及目录；目录于名称后加 "/", 可执行档于名称后加 "*" :
ls -AF

名称：more
使用权限：所有使用者
使用权限：所有使用者
使用方式：more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]
说明：类似 cat ，不过会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h 。
参数：-num 一次显示的行数
-d 提示使用者，在画面下方显示 [Press space to continue, q to quit.] ，如果使用者按错键，则会显示 [Press h for instructions.] 而不是 哔 声
-l 取消遇见特殊字元 ^L（送纸字元）时会暂停的功能
-f 计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）
-p 不以卷动的方式显示每一页，而是先清除萤幕后再显示内容
-c 跟 -p 相似，不同的是先显示内容再清除其他旧资料
-s 当遇到有连续两行以上的空白行，就代换为一行的空白行
-u 不显示下引号 （根据环境变数 TERM 指定的 terminal 而有所不同）
+/ 在每个档案显示前搜寻该字串（pattern），然后从该字串之后开始显示
+num 从第 num 行开始显示
fileNames 欲显示内容的档案，可为复数个数
范例：

more -s testfile 逐页显示 testfile 之档案内容，如有连续两行以上空白行则以一行空白行显示。
more +20 testfile 从第 20 行开始显示 testfile 之档案内容。

名称：mv
使用权限：所有使用者
使用方式：
mv [options] source dest
mv [options] source... directory
说明：将一个档案移至另一档案，或将数个档案移至另一目录。
参数：-i 若目的地已有同名档案，则先询问是否覆盖旧档。
范例：
将档案 aaa 更名为 bbb :
mv aaa bbb
将所有的C语言程式移至 Finished 子目录中 :
mv -i *.c

名称：rm
使用权限：所有使用者
使用方式：rm [options] name...
说明：删除档案及目录。
把计
-i 删除前逐一询问确认。
-f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。
-r 将目录及以下之档案亦逐一删除。
范例：
删除所有C语言程式档；删除前逐一询问确认 :
rm -i *.c
rm -i *.c
将 Finished 子目录及子目录中所有档案删除 :
rm -r Finished

名称：rmdir
使用权限：于目前目录有适当权限的所有使用者
使用方式： rmdir [-p] dirName
说明： 删除空的目录。
参数： -p 是当子目录被删除后使它也成为空目录的话，则顺便一并删除。
范例：
将工作目录下，名为 AAA 的子目录删除 :
rmdir AAA
在工作目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为空目录，则 BBB 亦予删除。
rmdir -p BBB/Test

名称：split
使用权限：所有使用者
使用方式：split [OPTION] [INPUT [PREFIX]]

说明：
将一个档案分割成数个。而从 INPUT 分割输出成固定大小的档案，其档名依序为 PREFIXaa, PREFIXab...；PREFIX 预设值为 `x。若没有 INPUT 档或为 `-，则从标准输入读进资料。
匡兜
-b, --bytes=SIZE
-b, --bytes=SIZE
SIZE 值为每一输出档案的大小，单位为 byte。
-C, --line-bytes=SIZE
每一输出档中，单行的最大 byte 数。
-l, --lines=NUMBER
NUMBER 值为每一输出档的列数大小。
-NUMBER
与 -l NUMBER 相同。
--verbose
于每个输出档被开启前，列印出侦错资讯到标准错误输出。
--help
显示辅助资讯然后离开。
--version
列出版本资讯然后离开。
SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。
范例：
PostgresSQL 大型资料库备份与回存：
因 Postgres 允许表格大过你系统档案的最大容量，所以要将表格 dump 到单一的档案可能会有问题，使用 split进行档案分割。
% pg_dump dbname | split -b 1m - filename.dump.
重新载入
% createdb dbname
% cat filename.dump.* | pgsql dbname
% cat filename.dump.* | pgsql dbname

名称：touch
使用权限：所有使用者
使用方式：
touch [-acfm]
[-r reference-file] [--file=reference-file]
[-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version]
file1 [file2 ...]
说明：
touch 指令改变档案的时间记录。 ls -l 可以显示档案的时间记录。
参数：
a 改变档案的读取时间记录。
m 改变档案的修改时间记录。
c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。
f 不使用，是为了与其他 unix 系统的相容性而保留。

r 使用参考档的时间记录，与 --file 的效果一样。
d 设定时间与日期，可以使用各种不同的格式。
t 设定档案的时间记录，格式与 date 指令相同。
--no-create 不会建立新档案。
--help 列出指令格式。
--version 列出版本讯息。
--version 列出版本讯息。
范例：
最简单的使用方式，将档案的时候记录改为现在的时间。若档案不存在，系统会建立一个新的档案。
touch file
touch file1 file2
将 file 的时间记录改为 5 月 6 日 18 点 3 分，公元两千年。时间的格式可以参考 date 指令，至少需输入 MMDDHHmm ，就是月日时与分。
touch -c -t 05061803 file
touch -c -t 050618032000 file
将 file 的时间记录改变成与 referencefile 一样。
touch -r referencefile file
将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是 24 小时的格式，日期可以使用其他格式如 6 May 2000 。
touch -d "6:03pm" file
touch -d "05/06/2000" file
touch -d "6:03pm 05/06/2000" file

名称 : at
使用权限 : 所有使用者
使用方式 : at -V [-q queue] [-f file] [-mldbv] TIME
说明 : at 可以让使用者指定在 TIME 这个特定时刻执行某个程式或指令，TIME 的格式是 HH:MM其中的 HH 为小时，MM 为分钟，甚至你也可以指定 am, pm, midnight, noon, teatime(就是下午 4 点锺)等口语词。
midnight, noon, teatime(就是下午 4 点锺)等口语词。
如果想要指定超过一天内的时间，则可以用 MMDDYY 或者 MM/DD/YY 的格式，其中 MM 是分钟，DD 是第几日，YY 是指年份。另外，使用者甚至也可以使用像是 now + 时间间隔来弹性指定时间，其中的时间间隔可以是 minutes, hours, days, weeks
另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter 之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作，至于执行的结果将会寄回你的帐号中。
把计 :
-V : 印出版本编号
-q : 使用指定的伫列(Queue)来储存，at 的资料是存放在所谓的 queue 中，使用者可以同时使用多个 queue，而 queue 的编号为 a, b, c... z 以及 A, B, ... Z 共 52 个
-m : 即使程式/指令执行完成后没有输出结果, 也要寄封信给使用者
-f file : 读入预先写好的命令档。使用者不一定要使用交谈模式来输入，可以先将所有的指定先写入档案后再一次读入
-l : 列出所有的指定 (使用者也可以直接使用 atq 而不用 at -l)

-d : 删除指定 (使用者也可以直接使用 atrm 而不用 at -d)
-v : 列出所有已经完成但尚未删除的指定
例子 :
三天后的下午 5 点钟执行 /bin/ls :
at 5pm + 3 days /bin/ls
三个星期后的下午 5 点钟执行 /bin/ls :
三个星期后的下午 5 点钟执行 /bin/ls :
at 5pm + 2 weeks /bin/ls
明天的 17:20 执行 /bin/date :
at 17:20 tomorrow /bin/date
1999 年的最后一天的最后一分钟印出 the end of world !
at 23:59 12/31/1999 echo the end of world !

名称：cal
使用权限：所有使用者
使用方式：cal [-mjy] [month [year]]
说明：
显示日历。若只有一个参数，则代表年份(1-9999)，显示该年的年历。年份必须全部写出：``cal 89 将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。
1752 年 9 月第 3 日起改用西洋新历，因这时大部份的国家都采用新历，有 10 天被去除，所以该月份的月历有些不同。在此之前为西洋旧历。
匡兜
-m : 以星期一为每周的第一天方式显示。
-j : 以凯撒历显示，即以一月一日起的天数显示。
-y : 显示今年年历。
范例：
cal : 显示本月的月历。
[root@mylinux /root]# date
Tue Aug 15 08:00:18 CST 2000
Tue Aug 15 08:00:18 CST 2000
[root@mylinux /root]# cal
August 2000
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
[root@mylinux /root]#
cal 2001 : 显示公元 2001 年年历。
[root@mylinux /root]# cal 2001
2001
January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1 2 3
7 8 9 10 11 12 13 4 5 6 7 8 9 10 4 5 6 7 8 9 10

14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17
21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24
28 29 30 31 25 26 27 28 25 26 27 28 29 30 31
April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 5 1 2
1 2 3 4 5 6 7 1 2 3 4 5 1 2
8 9 10 11 12 13 14 6 7 8 9 10 11 12 3 4 5 6 7 8 9
15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16
22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23
29 30 27 28 29 30 31 24 25 26 27 28 29 30
July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 1
8 9 10 11 12 13 14 5 6 7 8 9 10 11 2 3 4 5 6 7 8
15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15
22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22
29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29
30
October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1
7 8 9 10 11 12 13 4 5 6 7 8 9 10 2 3 4 5 6 7 8
14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15
21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22
28 29 30 31 25 26 27 28 29 30 23 24 25 26 27 28 29
30 31
[root@mylinux /root]#
cal 5 2001 : 显示公元 2001 年 5 月月历。
cal 5 2001 : 显示公元 2001 年 5 月月历。
[root@mylinux /root]# cal 5 2001
May 2001
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
[root@mylinux /root]#
cal -m : 以星期一为每周的第一天方式，显示本月的月历。
[root@mylinux /root]# cal -m
August 2000
Mo Tu We Th Fr Sa Su
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27

28 29 30 31

[root@mylinux /root]#

cal -jy : 以一月一日起的天数显示今年的年历。

[root@mylinux /root]# cal -jy

2000

2000

January February

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

1 32 33 34 35 36

2 3 4 5 6 7 8 37 38 39 40 41 42 43

9 10 11 12 13 14 15 44 45 46 47 48 49 50

16 17 18 19 20 21 22 51 52 53 54 55 56 57

23 24 25 26 27 28 29 58 59 60

30 31

March April

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

61 62 63 64 92

65 66 67 68 69 70 71 93 94 95 96 97 98 99

72 73 74 75 76 77 78 100 101 102 103 104 105 106

79 80 81 82 83 84 85 107 108 109 110 111 112 113

86 87 88 89 90 91 114 115 116 117 118 119 120

121

May June

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

122 123 124 125 126 127 153 154 155

128 129 130 131 132 133 134 156 157 158 159 160 161 162

135 136 137 138 139 140 141 163 164 165 166 167 168 169

142 143 144 145 146 147 148 170 171 172 173 174 175 176

142 143 144 145 146 147 148 170 171 172 173 174 175 176

149 150 151 152 177 178 179 180 181 182

July August

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

183 214 215 216 217 218

184 185 186 187 188 189 190 219 220 221 222 223 224 225

191 192 193 194 195 196 197 226 227 228 229 230 231 232

198 199 200 201 202 203 204 233 234 235 236 237 238 239

205 206 207 208 209 210 211 240 241 242 243 244

212 213

September October

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

245 246 275 276 277 278 279 280 281

247 248 249 250 251 252 253 282 283 284 285 286 287 288

254 255 256 257 258 259 260 289 290 291 292 293 294 295

261 262 263 264 265 266 267 296 297 298 299 300 301 302

268 269 270 271 272 273 274 303 304 305

November December

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat

306 307 308 309 336 337
310 311 312 313 314 315 316 338 339 340 341 342 343 344
317 318 319 320 321 322 323 345 346 347 348 349 350 351
324 325 326 327 328 329 330 352 353 354 355 356 357 358
324 325 326 327 328 329 330 352 353 354 355 356 357 358
331 332 333 334 335 359 360 361 362 363 364 365
366

名称 : crontab
使用权限 : 所有使用者
使用方式 :
crontab [-u user] filecrontab [-u user] { -l | -r | -e }
说明 :
crontab 是用来让使用者在固定时间或固定间隔执行程式之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。
餐数 :
-e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说 setenv VISUAL joe)
-r : 删除目前的时程表
-l : 列出目前的时程表
时程表的格式如下 :
f1 f2 f3 f4 f5 program
其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程式。
，f5 表示一个星期中的第几天。program 表示要执行的程式。
当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程式，其余类推
当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推
当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间间隔执行一次，其余类推
当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其余类推
使用者也可以将所有的设定先存放在档案 file 中，用 crontab file 的方式来设定时程表。
例子 :
每月每天每小时的第 0 分钟执行一次 /bin/ls :
0 7 * * * /bin/ls
在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup :
0 6-12/3 * 12 * /usr/bin/backup
周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata
每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"

20 0-23/2 * * * echo "haha"
注意 :
当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容
当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容
，若是你不希望收到这样的信，请在每一行空一格之后加上 > /dev/null 2>&1
即可。

名称 : date
使用权限 : 所有使用者
使用方式 :
date [-u] [-d datestr] [-s datestr] [--utc] [--universal]
[--date=datestr] [--set=datestr] [--help] [--version] [+FORMAT]
[MMDDhhmm[[CC]YY][.ss]]
说明 :
date 可以用来显示或设定系统的日期与时间，在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中可用的标记列表如下 :
时间方面 :
% : 印出 %
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM
%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%r : 直接显示时间 (12 小时制，格式为 hh:mm:ss [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数
%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区
日期方面 :
%a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)
%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)
%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)

%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)
%Y : 完整年份 (0000..9999)
若是不以加号作为开头，则表示要设定时间，而时间格式为
MMDDhhmm[[CC]YY][.ss]，其中 MM 为月份，DD 为日，hh 为小时，mm 为分钟
，CC 为年份前两位数字，YY 为年份后两位数字，ss 为秒数
把计 :
-d datestr : 显示 datestr 中所设定的时间 (非系统时间)
--help : 显示辅助讯息
-s datestr : 将系统时间设为 datestr 中所设定的时间
-u : 显示目前的格林威治时间
--version : 显示版本编号
例子 :
显示时间后跳行，再显示目前日期 :
date +%T%n%D
显示月份与日数 :
date +%B %d
显示日期与设定时间(12:34:56) :
date --date 12:34:56
注意 :
当你不希望出现无意义的 0 时(比如说 1999/03/07)，则可以在标记中插入 - 符号，比如说 date +%-H:%-M:%-S 会把时分秒中无意义的 0 给去掉，像是原本的 08:09:04 会变为 8:9:4。另外，只有取得权限者(比如说 root)才能设定系统时间。
当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入
当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入
CMOS 中，这样下次重新开机时系统时间才会持续抱持最新的正确值。

名称 : sleep
使用权限 : 所有使用者
使用方式 : sleep [--help] [--version] number[smhd]
说明 : sleep 可以用来将目前动作延迟一段时间
参数说明 :
--help : 显示辅助讯息
--version : 显示版本编号
number : 时间长度，后面可接 s、m、h 或 d
其中 s 为秒，m 为 分钟，h 为小时，d 为日数
例子 :
显示目前时间后延迟 1 分钟，之后再次显示时间 :
date;sleep 1m;date

名称： time
使用权限： 所有使用者
使用方式： time [options] COMMAND [arguments]
说明： time 指令的用途，在于量测特定指令执行时所需消耗的时间及系统资源等资讯。例如 CPU 时间、记忆体、输入输出等等。需要特别注意的是，部分资讯

在 Linux 上显示不出来。这是因为在 Linux 上部分资源的分配函式与 time 指令所预设的方式并不相同，以致于 time 指令无法取得这些资料。
把计
-o or --output=FILE
-o or --output=FILE
设定结果输出档。这个选项会将 time 的输出写入 所指定的档案中。如果档案已经存在，系统将覆写其内容。
-a or --append
配合 -o 使用，会将结果写到档案的末端，而不会覆盖掉原来的内容。
-f FORMAT or --format=FORMAT
以 FORMAT 字串设定显示方式。当这个选项没有被设定的时候，会用系统预设的格式。不过你可以用环境变数 time 来设定这个格式，如此一来就不必每次登入系统都要设定一次。
一般设定上，你可以用
t
表示跳栏，或者是用
n
表示换行。每一项资料要用 % 做为前导。如果要在字串中使用百分比符号，就用
。（学过 C 语言的人大概会觉得很熟悉）
time 指令可以显示的资源有四大项，分别是：
Time resources
Memory resources
IO resources
Command info
详细的内容如下：
Time Resources
E 执行指令所花费的时间，格式是：[hour]:minute:second。请注意这个数字并
E 执行指令所花费的时间，格式是：[hour]:minute:second。请注意这个数字并
不代表实际的 CPU 时间。
e 执行指令所花费的时间，单位是秒。请注意这个数字并不代表实际的 CPU 时间
。
S 指令执行时在核心模式（kernel mode）所花费的时间，单位是秒。
U 指令执行时在使用者模式（user mode）所花费的时间，单位是秒。
P 执行指令时 CPU 的占用比例。其实这个数字就是核心模式加上使用者模式的 CPU 时间除以总时间。
Memory Resources
M 执行时所占用的实体记忆体的最大值。单位是 KB
t 执行时所占用的实体记忆体的平均值，单位是 KB
K 执行程序所占用的记忆体总量（stack+data+text）的平均大小，单位是 KB
D 执行程序的自有资料区（unshared data area）的平均大小，单位是 KB
p 执行程序的自有堆叠（unshared stack）的平均大小，单位是 KB
X 执行程序间共享内容（shared text）的平均值，单位是 KB
Z 系统记忆体页的大小，单位是 byte。对同一个系统来说这是个常数
IO Resources
F 此程序的主要记忆体页错误发生次数。所谓的主要记忆体页错误是指某一记忆体页已经置换到置换档（swap file)中，而且已经分配给其他程序。此时该页的内容必须从置换档里再读出来。

R 此程序的次要记忆体页错误发生次数。所谓的次要记忆体页错误是指某一记忆体页虽然已经置换到置换档中，但尚未分配给其他程序。此时该页的内容并未被破坏，不必从置换档里读出来
破坏，不必从置换档里读出来
W 此程序被交换到置换档的次数
c 此程序被强迫中断（像是分配到的 CPU 时间耗尽）的次数
w 此程序自愿中断（像是在等待某一个 I/O 执行完毕，像是磁碟读取等等）的次数
I 此程序所输入的档案数
O 此程序所输出的档案数
r 此程序所收到的 Socket Message
s 此程序所送出的 Socket Message
k 此程序所收到的信号 (Signal)数量
Command Info
C 执行时的参数以及指令名称
x 指令的结束代码 (Exit Status)
-p or --portability
这个选项会自动把显示格式设定成为：
real %e
user %U
sys %S
这么做的目的是为了与 POSIX 规格相容。
-v or --verbose
这个选项会把所有程式中用到的资源通通列出来，不但如一般英文语句，还有说明。对不想花时间去熟习格式设定或是刚刚开始接触这个指令的人相当有用。
范例：
范例：
利用下面的指令
time -v ps -aux
我们可以获得执行 ps -aux 的结果和所花费的系统资源。如下面所列的资料：
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 1096 472 ? S Apr19 0:04 init
root 2 0.0 0.0 0 0 ? SW Apr19 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Apr19 0:00 [kpiod]
......
root 24269 0.0 1.0 2692 996 pts/3 R 12:16 0:00 ps -aux
Command being timed: "ps -aux"
User time (seconds): 0.05
System time (seconds): 0.06
Percent of CPU this job got: 68%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 0
Average resident set size (kbytes): 0

Major (requiring I/O) page faults: 238
Minor (reclaiming a frame) page faults: 46
Minor (reclaiming a frame) page faults: 46
Voluntary context switches: 0
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
名称： uptime
使用权限： 所有使用者

使用方式： uptime [-V]
说明： uptime 提供使用者下面的资讯，不需其他参数：
现在的时间
系统开机运转到现在经过的时间
连线的使用者数量
最近一分钟，五分钟和十五分钟的系统负载
参数： -V 显示版本资讯。
范例： uptime
其结果为：
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99

名称：chfn
使用权限：所有使用者
用法：shell>> chfn
说明：提供使用者更改个人资讯，用于 finger and mail uname
范例：
shell>> chfn
Changing finger information for user
Password: [del]
Name[]:Johnney Huang ### 提供 finger 时的资料
Office[]:NCCU
Office Phone[]: [del]
Home Phone[]: [del]

名称：chsh
使用权限：所有使用者
用法：shell>> chsh
说明：更改使用者 shell 设定
范例：
shell>> chsh

Changing fihanging shell for user1
Password: [del]
New shell [/bin/tcsh]: ### [是目前使用的 shell]
[del]
[del]
shell>> chsh -l ### 展示 /etc/shells 档案内容
/bin/bash
/bin/sh
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh

名称： finger
使用权限： 所有使用者
使用方式： finger [options] user[@address]
说明：finger 可以让使用者查询一些其他使用者的资料。会列出来的资料有：
Login Name
User Name
Home directory
Shell
Login status
mail status
.plan
.project
.forward
其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的
其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的
.plan ， .project 和 .forward 等档案里的资料。如果没有就没有。finger 指令并不限定于在同一伺服器上查询，也可以寻找某一个远端伺服器上的使用者。只要给一个像是 E-mail address 一般的地址即可。
把计
-l
多行显示。
-s
单行显示。这个选项只显示登入名称，真实姓名，终端机名称，闲置时间，登入时间，办公室号码及电话号码。如果所查询的使用者是远端伺服器的使用者，这个选项无效。
范例：下列指令可以查询本机管理员的资料：
finger root
其结果如下：
Login: root Name: root
Directory: /root Shell: /bin/bash
Never logged in.
No mail.
No Plan.

名称：last
使用权限：所有使用者
使用方式：shell>> last [options]
说明：显示系统开机以来获是从每月初登入者的讯息
说明：显示系统开机以来获是从每月初登入者的讯息
把计
-R 省略 hostname 的栏位
-num 展示前 num 个
uname 展示 uname 的登入讯息
tty 限制登入讯息包含终端机代号
范例：
shell>> last -R -2
johnney pts/1 Mon Aug 14 20:42 still logged in
johnney pts/0 Mon Aug 14 19:59 still logged in
wtmp begins Tue Aug 1 09:01:10 2000 ### /var/log/wtmp
shell>> last -2 minery
minery pts/0 140.119.217.115 Mon Aug 14 18:37 - 18:40 (00:03)
minery pts/0 140.119.217.115 Mon Aug 14 17:22 - 17:24 (00:02)
wtmp begins Tue Aug 1 09:01:10 2000
名称:login
这个命令都不会就不要干算了！呵呵我也不在这里多费笔墨耽误大家美好青春了^_^

名称：passwd
使用权限：所有使用者
使用方式：passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]
说明：用来更改使用者的密码
参数：
参数：
-k
-l
-u
-f
-d 关闭使用者的密码认证功能, 使用者在登入时将可以不用输入密码, 只有具备 root 权限的使用者方可使用.
-S 显示指定使用者的密码认证种类, 只有具备 root 权限的使用者方可使用.
[username] 指定帐号名称.

名称 : who
使用权线 : 所有使用者都可使用
使用方式 : who - [husfV] [user]
说明 : 显示系统中有那些使用者正在上面，显示的资料包含了使用者 ID，使用的终端机，从那边连上来的，上线时间，呆滞时间，CPU 使用量，动作等等。
把计 :
-h : 不要显示标题列
-u : 不要显示使用者的动作/工作
-s : 使用简短的格式来显示

-f ： 不要显示使用者的上线位置
-V ： 显示程式版本

名称：/etc/aliases
使用权限：系统管理者
使用方式： 请用 newaliases 更新资料库
说明：
sendmail 会使用一个在 /etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据 aliases档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在 mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫 redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。
/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。newaliases 的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。范例：
newaliases
下面命令会做相同的事，
sendmail -bi
相关命令:
mail, mailq, newaliases, sendmail

名称：mail
使用权限：所有使用者
使用方式：mail [-iInv] [-s subject] [-c cc-addr] [-b bcc-addr] user1 [user 2 ...]
说明：
说明：
mail 不仅只是一个指令， mail 还是一个电子邮件程式，不过利用 mail 来读信的人应该很少吧！对于系统管理者来说 mail 就很有用，因为管理者可以用 mail 写成 script ，定期寄一些备忘录提醒系统的使用者。
参数：
i 忽略 tty 的中断讯号。 (interrupt)
I 强迫设成互动模式。 (Interactive)
v 列印出讯息，例如送信的地点、状态等等。 (verbose)
n 不读入 mail.rc 设定档。
s 邮件标题。
c cc 邮件地址。
b bcc 邮件地址。
范例：
将信件送给一个或以上的电子邮件地址，由于没有加入其他的选项，使用者必须输入标题与信件的内容等。而 user2 没有主机位置，就会送给邮件伺服器的 user2 使用者。
mail user1@email.address
mail user1@email.address user2
将 mail.txt 的内容寄给 user2 同时 cc 给 user1 。如果将这一行指令设成

cronjob 就可以定时将备忘录寄给系统使用者。
mail -s 标题 -c user1 user2 < mail.txt

指令：mesg
使用权限 : 所有使用者
使用权限 : 所有使用者
使用方式 : mesg [y|n]
说明 : 决定是否允许其他人传讯息到自己的终端机介面
把计
y : 允许讯息传到终端机介面上。
n : 不允许讯息传到终端机介面上 。
如果没有设定，则讯息传递与否则由终端机界面目前状态而定。
例子 :
改变目前讯息设定，改成不允许讯息传到终端机介面上 :
mesg n
与 mesg 相关的指令有： talk，write，wall。

名称 : talk
使用权限 : 所有使用者
使用方式 :
talk person [ttyname]
说明 : 与其他使用者对谈
把计 :
person : 预备对谈的使用者帐号，如果该使用者在其他机器上，则可输入 person@machine.name
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息
例子.1 :
与现在机器上的使用者Rollaend对谈，此时 Rollaend 只有一个连线 :
talk Rollaend
接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 `talk jzlee`即
接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 `talk jzlee`即
可开始对谈，结束请按 ctrl+c
例子.2 :与linuxfab.cx上的使用者Rollaend对谈，使用pts/2来对谈 :
talk Rollaend@linuxfab.cx pts/2
接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 `talk jzlee@jzlee.home`即可开始对谈，结束请按 ctrl+c
注意 : 若萤幕的字会出现不正常的字元，试着按 ctrl+l 更新萤幕画面。

名称 : wall
使用权限 : 所有使用者
使用方式 :
wall [message]
使用说明：
wall 会将讯息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端机介面做为标准传入时, 讯息结束时需加上 EOF (通常用 Ctrl+D)

例子 :
传讯息"hi" 给每一个使用者 :
wall hi
名称 : write
使用权限 : 所有使用者
使用方式 :
write user [ttyname]
说明 : 传讯息给其他使用者
把计 :
把计 :
user : 预备传讯息的使用者帐号
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息
例子.1 :
传讯息给 Rollaend，此时 Rollaend 只有一个连线 :
write Rollaend
接下来就是将讯息打上去，结束请按 ctrl+c
例子.2 :传讯息给 Rollaend，Rollaend 的连线有 pts/2，pts/3 :
write Rollaend pts/2
接下来就是将讯息打上去，结束请按 ctrl+c
注意 : 若对方设定 mesg n，则此时讯席将无法传给对方

名称：kill
使用权限：所有使用者
使用方式：
kill [-s signal | -p] [-a] pid ...
kill -l [signal]
说明：kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作, 若没有指定, 预设是送出终止 (TERM) 的信号
把计
-s (signal) : 其中可用的讯号有 HUP (1), KILL (9), TERM (15), 分别代表着重跑, 砍掉, 结束; 详细的信号可以用 kill -l
-p : 印出 pid , 并不送出信号
-p : 印出 pid , 并不送出信号
-l (signal) : 列出所有可用的信号名称
范例：
将 pid 为 323 的行程砍掉 (kill) :
kill -9 323
将 pid 为 456 的行程重跑 (restart) :
kill -HUP 456

名称：nice
使用权限：所有使用者
使用方式：nice [-n adjustment] [-adjustment] [--adjustment=adjustment] [--help] [--version] [command [arg...]]
说明：以更改过的优先序来执行程式, 如果未指定程式, 则会印出目前的排程优先序, 内定的 adjustment 为 10, 范围为 -20 (最高优先序) 到 19 (最低优先

序)
把计
-n adjustment, -adjustment, --adjustment=adjustment 皆为将该原有优先序的增加 adjustment
--help 显示求助讯息
--version 显示版本资讯
范例：
将 ls 的优先序加 1 并执行 :
nice -n 1 ls
将 ls 的优先序加 10 并执行 :
将 ls 的优先序加 10 并执行 :
nice ls将 ls 的优先序加 10 并执行
注意 : 优先序 (priority) 为作业系统用来决定 CPU 分配的参数，Linux 使用『回合制(round-robin)』的演算法来做 CPU 排程，优先序越高，所可能获得的CPU时间就越多。

名称：ps
使用权限：所有使用者
使用方式：ps [options] [--help]
说明：显示瞬间行程 (process) 的动态
参数：
ps 的参数非常多, 在此仅列出几个常用的参数并大略介绍含义
-A 列出所有的行程
-w 显示加宽可以显示较多的资讯
-au 显示较详细的资讯
-aux 显示所有包含其他使用者的行程
au(x) 输出格式 :
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
USER: 行程拥有者
PID: pid
%CPU: 占用的 CPU 使用率
%MEM: 占用的记忆体使用率
VSZ: 占用的虚拟记忆体大小
RSS: 占用的记忆体大小
RSS: 占用的记忆体大小
TTY: 终端的次要装置号码 (minor device number of tty)
STAT: 该行程的状态:
D: 不可中断的静止 (通悸□□缜b进行 I/O 动作)
R: 正在执行中
S: 静止状态
T: 暂停执行
Z: 不存在但暂时无法消除
W: 没有足够的记忆体分页可分配
N: 低优先序的行程
L: 有记忆体分页分配并锁在记忆体内 (即时系统或捱A I/O)
START: 行程开始时间
TIME: 执行的时间

COMMAND:所执行的指令
范例：
ps
PID TTY TIME CMD
2791 ttyp0 00:00:00 tcsh
3092 ttyp0 00:00:00 ps
% ps -A
PID TTY TIME CMD
1 ? 00:00:03 init
2 ? 00:00:00 kflushd
2 ? 00:00:00 kflushd
3 ? 00:00:00 kpiod
4 ? 00:00:00 kswapd
5 ? 00:00:00 mdrecoveryd
....
% ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.7 1096 472 ? S Sep10 0:03 init [3]
root 2 0.0 0.0 0 0 ? SW Sep10 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Sep10 0:00 [kpiod]
root 4 0.0 0.0 0 0 ? SW Sep10 0:00 [kswapd]
....

名称：pstree
使用权限：所有使用者
使用方式：
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-G|-U] [pid|user]
pstree -V
说明：将所有行程以树状图显示，树状图将会以 pid (如果有指定) 或是以
init 这个基本行程为根 (root)，如果有指定使用者 id ，则树状图会只显示该
使用者所拥有的行程
参数：
-a 显示该行程的完整指令及参数，如果是被记忆体置换出去的行程则会加上括号
-c 如果有重覆的行程名，则分开列出（预设值是会在前面加上 *
-c 如果有重覆的行程名，则分开列出（预设值是会在前面加上 *
范例：
pstree
init-+-amd
|-apmd
|-atd
|-httpd---10*[httpd]
%pstree -p
init(1)-+-amd(447)
|-apmd(105)
|-atd(339)
%pstree -c
init-+-amd

```
|-apmd
|-atd
|-httpd-+-httpd
| |-httpd
| |-httpd
| |-httpd
....
```

名称：renice
使用权限：所有使用者
使用方式：renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user
使用方式：renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user
...]
说明：重新指定一个或多个行程(Process)的优先序(一个或多个将根据所下的参数而定)
把计
-p pid 重新指定行程的 id 为 pid 的行程的优先序
-g pgrp 重新指定行程群组(process group)的 id 为 pgrp 的行程 (一个或多个) 的优先序
-u user 重新指定行程拥有者为 user 的行程的优先序
范例：
将行程 id 为 987 及 32 的行程与行程拥有者为 daemon 及 root 的优先序号码加 1 :
renice +1 987 -u daemon root -p 32
注意 : 每一个行程(Process)都有一个唯一的 (unique) id

名称：top
使用权限：所有使用者
使用方式：top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]
说明：即时显示 process 的动态
把计
d : 改变显示的更新速度，或是在交谈式指令列(interactive command)按 s
q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完
c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称S : 累积模式，会将己完成或消失的子行程 (dead child process) 的 CPU time 累积起来
s : 安全模式，将交谈式指令取消, 避免潜在的危机
i : 不显示任何闲置 (idle) 或无用 (zombie) 的行程
n : 更新的次数，完成后将会退出 top
b : 批次档模式，搭配 "n" 参数一起使用，可以用来将 top 的结果输出到档案内
范例：
显示更新十次后退出 ;
top -n 10
使用者将不能利用交谈式指令来对行程下命令 :

top -s
将更新显示二次的结果输入到名称为 top.log 的档案里 :
top -n 2 -b < top.log

名称：skill
使用权限：所有使用者
使用方式： skill [signal to send] [options] 选择程序的规则
说明：
送个讯号给正在执行的程序,预设的讯息为 TERM (中断) , 较常使用的讯息为
HUP , INT , KILL , STOP , CONT ,和 0
讯息有三种写法:分别为 -9 , -SIGKILL , -KILL , 可以使用 -l 或 -L 已列出
可使用的讯息。
可使用的讯息。
一般参数：
-f 快速模式/尚未完成
-i 互动模式/ 每个动作将要被确认
-v 详细输出/ 列出所选择程序的资讯
-w 智能警告讯息/ 尚未完成
-n 没有动作/ 显示程序代号
参数：选择程序的规则可以是, 终端机代号,使用者名称,程序代号,命令名称。
-t 终端机代号 (tty 或 pty)
-u 使用者名称
-p 程序代号 (pid)
-c 命令名称 可使用的讯号:
以下列出已知的讯号名称,讯号代号,功能。
名称 (代号) 功能/ 描述
ALRM 14 离开
HUP 1 离开
INT 2 离开
KILL 9 离开/ 强迫关闭
PIPE 13 离开
POLL 离开
PROF 离开
TERM 15 离开
USR1 离开
USR1 离开
USR2 离开
VTALRM 离开
STKFLT 离开/ 只适用于i386, m68k, arm 和 ppc 硬体
UNUSED 离开/ 只适用于i386, m68k, arm 和 ppc 硬体
TSTP 停止 /产生与内容相关的行为
TTIN 停止 /产生与内容相关的行为
TTOU 停止 /产生与内容相关的行为
STOP 停止 /强迫关闭
CONT 从新启动 /如果在停止状态则从新启动,否则忽略
PWR 忽略 /在某些系统中会离开
WINCH 忽略

CHLD 忽略
ABRT 6 核心
FPE 8 核心
ILL 4 核心
QUIT 3 核心
SEGV 11 核心
TRAP 5 核心
SYS 核心 /或许尚未实作
EMT 核心 /或许尚未实作
BUS 核心 /核心失败
XCPU 核心 /核心失败
XCPU 核心 /核心失败
XFSZ 核心 /核心失败
范例：
停止所有在 PTY 装置上的程序
skill -KILL -v pts/*
停止三个使用者 user1 , user2 , user3
skill -STOP user1 user2 user3
其他相关的命令: kill

名称：expr
使用权限：所有使用者
字串长度
shell>> expr length "this is a test"
14
数字商数
shell>> expr 14 % 9
5
从位置处抓取字串
shell>> expr substr "this is a test" 3 5
is is
数字串 only the first character
shell>> expr index "testforthegame" e
2
字串真实重现
字串真实重现
shell>> expr quote thisisatestformela
thisisatestformela

名称: tr
1.比方说要把目录下所有的大写档名换为小写档名?
似乎有很多方式，"tr"是其中一种:
```
#!/bin/sh
dir="/tmp/testdir";
files=`find $dir -type f`;
for i in $files
do
```

dir_name=`dirname $i`;
ori_filename=`basename $i`
new_filename=`echo $ori_filename | tr [:upper:] [:lower:]` >
/dev/null;
#echo $new_filename;
mv $dir_name/$ori_filename $dir_name/$new_filename
done
2.自己试验中...lowercase to uppercase
tr abcdef...[del] ABCDE...[del]
tr a-z A-Z
tr [:lower:] [:upper:]
shell>> echo "this is a test" | tr a-z A-Z > www
shell>> echo "this is a test" | tr a-z A-Z > www
shell>> cat www
THIS IS A TEST
3.去掉不想要的字串
shell>> tr -d this ### 去掉有关 t.e.s.t
this
man
man
test
e
4.取代字串
shell>> tr -s "this" "TEST"
this
TEST
th
TE

指令：clear
用途：清除萤幕用。
使用方法：在 console 上输入 clear。

名称: reset, tset
使用方法: tset [-IQqrs] [-] [-e ch] [-i ch] [-k ch] [-m mapping]
[terminal]
使用说明:
使用说明:
reset 其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是 ? 的话，这个程式会要求使用者输入终端机的型别。
由于这个程式会将终端机设回原始的状态，除了在 login 时使用外，当系统终端机因为程式不正常执行而进入一些奇怪的状态时，你也可以用它来重设终端机o 例如不小心把二进位档用 cat 指令进到终端机，常会有终端机不再回应键盘输入，或是回应一些奇怪字元的问题。此时就可以用 reset 将终端机回复至原始状态。选项说明:
-p

将终端机类别显示在萤幕上，但不做设定的动作。这个命令可以用来取得目前终端机的类别。
-e ch
将 erase 字元设成 ch
-i ch
将中断字元设成 ch
-k ch
将删除一行的字元设成 ch
-I
不要做设定的动作，如果没有使用选项 -Q 的话，erase、中断及删除字元的目前值依然会送到萤幕上。
-Q
不要显示 erase、中断及删除字元的值到萤幕上。
不要显示 erase、中断及删除字元的值到萤幕上。
-r
将终端机类别印在萤幕上。
-s
将设定 TERM 用的命令用字串的型式送到终端机中，通常在 .login 或
.profile 中用
范例:
让使用者输入一个终端机型别并将终端机设到该型别的预设状态。
reset ?
将 erase 字元设定 control-h
reset -e ^B
将设定用的字串显示在萤幕上
reset -s
Erase is control-B (^B).
Kill is control-U (^U).
Interrupt is control-C (^C).
TERM=xterm;

名称：compress
使用权限：所有使用者
使用方式：compress [-dfvcV] [-b maxbits] [file ...]
说明：
compress 是一个相当古老的 unix 档案压缩指令，压缩后的档案会加上一个 .Z 延伸档名以区别未压缩的档案，压缩后的档案可以以 uncompress 解压。若要将
延伸档名以区别未压缩的档案，压缩后的档案可以以 uncompress 解压。若要将
数个档案压成一个压缩档，必须先将档案 tar 起来再压缩。由于 gzip 可以产生更理想的压缩比例，一般人多已改用 gzip 为档案压缩工具。
参数：
c 输出结果至标准输出设备（一般指荧幕）
f 强迫写入档案，若目的档已经存在，则会被覆盖 (force)
v 将程式执行的讯息印在荧幕上 (verbose)
b 设定共同字串数的上限，以位元计算，可以设定的值为 9 至 16 bits 。由于值越大，能使用的共同字串就 越多，压缩比例就越大，所以一般使用预设值 16 bits (bits)

d 将压缩档解压缩
V 列出版本讯息
范例：
将 source.dat 压缩成 source.dat.Z ，若 source.dat.Z 已经存在，内容则会被压缩档覆盖。
compress -f source.dat
将 source.dat 压缩成 source.dat.Z ，并列印出压缩比例。
-v 与 -f 可以一起使用
compress -vf source.dat
将压缩后的资料输出后再导入 target.dat.Z 可以改变压缩档名。
compress -c source.dat > target.dat.Z
-b 的值越大，压缩比例就越大，范围是 9-16 ，预设值是 16 。
compress -b 12 source.dat
compress -b 12 source.dat
将 source.dat.Z 解压成 source.dat ，若档案已经存在，使用者按 y 以确定覆盖档案，若使用 -df 程式则会自动覆盖档案。由于系统会自动加入 .Z 为延伸档名，所以 source.dat 会自动当作 source.dat.Z 处理。
compress -d source.dat
compress -d source.dat.Z

名称： lpd
使用权限： 所有使用者
使用方式：lpd [-l] [#port]
lpd 是一个常驻的印表机管理程式，它会根据 /etc/printcap 的内容来管理本地或远端的印表机。/etc/printcap 中定义的每一个印表机必须在 /var/lpd 中有一个相对应的目录，目录中以 cf 开头的档案表示一个等待送到适当装置的印表工作。这个档案通常是由 lpr 所产生。
lpr 和 lpd 组成了一个可以离线工作的系统，当你使用 lpr 时，印表机不需要能立即可用，甚至不用存在。lpd 会自动监视印表机的状况，当印表机上线后，便立即将档案送交处理。这个得所有的应用程式不必等待印表机完成前一工作。
参数：
-l: 将一些除错讯息显示在标准输出上。
#port: 一般而言，lpd 会使用 getservbyname 取得适当的 TCP/IP port，你可以使用这个参数强迫 lpd 使用指定的 port。
范例：
这个程式通常是由 /etc/rc.d 中的程式在系统启始阶段执行。

名称 lpq
-- 显示列表机贮列中未完成的工作 用法
lpq [l] [P] [user]
说明
lpq 会显示由 lpd 所管理的列表机贮列中未完成的项目。
范例
范例 1. 显示所有在 lp 列表机贮列中的工作
lpq -PlpRank Owner Job Files Total Size1st root 238 (standard input) 1428646 bytes
相关函数

lpr,lpc,lpd

名称： lpr
使用权限： 所有使用者
使用方式：lpr [-P printer]
将档案或是由标准输入送进来的资料送到印表机贮列之中，印表机管理程式 lpd 会在稍后将这个档案送给适当的程式或装置处理。lpr 可以用来将料资送给本地或是远端的主机来处理。
参数：
-p Printer: 将资料送至指定的印表机 Printer，预设值为 lp。
范例：
将 http://www.c/和 kkk.c 送到印表机 lp。
lpr -Plp http://www.c/kkk.c
名称: lprm
名称: lprm
-- 将一个工作由印表机贮列中移除 用法
/usr/bin/lprm [P] [file...]
说明
尚未完成的印表机工作会被放在印表机贮列之中，这个命令可用来将常未送到印表机的工作取消。由于每一个印表机都有一个独立的贮列，你可以用 -P 这个命令设定想要作用的印列机。如果没有设定的话，会使用系统预设的印表机。
这个命令会检查使用者是否有足够的权限删除指定的档案，一般而言，只有档案的拥有者或是系统管理员才有这个权限。
范例
将印表机 hpprinter 中的第 1123 号工作移除
lprm -Phpprinter 1123
将第 1011 号工作由预设印表机中移除
lprm 1011

名称： fdformat
使用权限： 所有使用者
使用方式：fdformat [-n] device
使用说明 :
对指定的软碟机装置进行低阶格式化。使用这个指令对软碟格式化的时候，最好指定像是下面的装置：
/dev/fd0d360 磁碟机 A: ，磁片为 360KB 磁碟
/dev/fd0h1440 磁碟机 A: ，磁片为 1.4MB 磁碟
/dev/fd1h1200 磁碟机 B: ，磁片为 1.2MB 磁碟
/dev/fd1h1200 磁碟机 B: ，磁片为 1.2MB 磁碟
如果使用像是 /dev/fd0 之类的装置，如果里面的磁碟不是标准容量，格式化可能会失败。在这种情况之下，使用者可以用 setfdprm 指令先行指定必要参数。
参数：
-n 关闭确认功能。这个选项会关闭格式化之后的确认步骤。
范例：
fdformat -n /dev/fd0h1440
将磁碟机 A 的磁片格式化成 1.4MB 的磁片。并且省略确认的步骤。

名称： mformat

使用权限： 所有使用者

使用方式：

mformat [-t cylinders] [-h heads] [-s sectors] [-l volume_label] [-F] [-I fsVer-sion] [-S sizecode] [-2 sectors_on_track_0] [-M software_sector_size] [-a] [-X] [-C] [-H hidden_sectors] [-r root_sectors] [-B boot_sector] [-0 rate_on_track_0] [-A rate_on_other_tracks] [-1] [-k] drive:

在已经做过低阶格式化的磁片上建立 DOS 档案系统。如果在编译 mtools 的时候把 USE_2M 的参数打开，部分与 2M 格式相关的参数就会发生作用。否则这些参数（像是 S,2,1,M）不会发生作用。

参数：

-t 磁柱（synlider）数

-h 磁头（head）数

-s 每一磁轨的磁区数

-s 每一磁轨的磁区数

-l 标签

-F 将磁碟格式化为 FAT32 格式，不过这个参数还在实验中。

-I 设定 FAT32 中的版本号。这当然也还在实验中。

-S 磁区大小代码，计算方式为 sector = 2^(大小代码+7)

-c 磁丛（cluster）的磁区数。如果所给定的数字会导致磁丛数超过 FAT 表的限制，mformat 会自动放大磁区数。

-s

-M 软体磁区大小。这个数字就是系统回报的磁区大小。通常是和实际的大小相同。

-a 如果加上这个参数，mformat 会产生一组 Atari 系统的序号给这块软碟。

-X 将软碟格式化成 XDF 格式。使用前必须先用 xdfcopy 指令对软碟作低阶格式化的动作。

-C 产生一个可以安装 MS-DOS 档案系统的磁碟影像档（disk image）。当然对一个实体磁碟机下这个参数是没有意义的。

-H 隐藏磁区的数目。这通常适用在格式化硬碟的分割区时，因为通常一个分割区的前面还有分割表。这个参数未经测试，能不用就不用。

-n 磁碟序号

-r 根目录的大小，单位是磁区数。这个参数只对 FAT12 和 FAT16 有效。

-B 使用所指定的档案或是设备的开机磁区做为这片磁片或分割区的开机磁区。当然当中的硬体参数会随之更动。

-k 尽量保持原有的开机磁区。

-0 第 0 轨的资料传输率

-0 第 0 轨的资料传输率

-A 第 0 轨以外的资料传输率

-2 使用 2m 格式

-1 不使用 2m 格式

范例：

mformat a:

这样会用预设值把 a: （就是 /dev/fd0）里的磁碟片格式化。

s 名称： mkdosfs

使用权限： 所有使用者
使用方式： mkdosfs [-c | -l filename]
[-f number_of_FATs]
[-F FAT_size]
[-i volume_id]
[-m message_file]
[-n volume_name]
[-r root_dir_entry]
[-s sector_per_cluster]
[-v]
device
[block_count]
说明： 建立 DOS 档案系统。 device 指你想要建立 DOS 档案系统的装置代号。像是 /dev/hda1 等等。 block_count 则是你希望配置的区块数。如果
block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。
block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。
参数：
-c 建立档案系统之前先检查是否有坏轨。
-l 从得定的档案中读取坏轨记录。
-f 指定档案配置表（FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 档案系统不支援超过 2 个 FAT 表。通常这个不需要改。
-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于磁碟片，16 位元组用于一般硬碟的分割区，也就是所谓的 FAT16 格式。这个值通常系统会自己选定适当的值。在磁碟片上用 FAT16 通常不会发生作用，反之在硬碟上用 FAT12 亦然。
-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。
-m 当使用者试图用这片磁片或是分割区开机，而上面没有作业系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用档案编辑好，然后用这个参数指定，或是用
-m -
这样系统会要求你直接输入这段文字。要特别注意的是，档案里的字串长度不要超过 418 个字，包括展开的跳栏符号（TAB）和换行符号（换行符号在 DOS 底下算两个字元！）
-n 指定 Volume Name，就是磁碟标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。
-r 指定根目录底下的最大档案数。这里所谓的档案数包括目录。预设值是在软碟上是 112 或是 224 ，在硬碟上是 512。没事不要改这个数字。
好，然后用这个参数指定，或是用
-s 每一个磁丛（cluster）的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。
-v 提供额外的讯息
范例：
mkdosfs -n Tester /dev/fd0 将 A 槽里的磁碟片格式化为 DOS 格式，并将标签设为 Tester

上一页 目录 下一页

linux指令大全

名称：cat
使用权限：所有使用者
使用方式：cat [-AbeEnstTuv] [--help] [--version] fileName
说明：把档案串连接后传到基本输出（萤幕或加 > fileName 到另一个档案）
参数：
-n 或 --number 由 1 开始对所有输出的行数编号
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行
-v 或 --show-nonprinting

范例：
cat -n textfile1 > textfile2 把 textfile1 的档案内容加上行号后输入 textfile2 这个档案里
cat -b textfile1 textfile2 >> textfile3 把 textfile1 和 textfile2 的档案内容加上行号（空白行不加）之后将内容附加到 textfile3

名称 : cd
使用权限 : 所有使用者

使用方式 : cd [dirName]

说明 : 变换工作目录至 dirName。 其中 dirName 表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的 home directory (也就是刚 login 时所在的目录)。

另外，"~" 也表示为 home directory 的意思，"." 则是表示目前所在的目录，".." 则表示目前目录位置的上一层目录。

范例 : 跳到 /usr/bin/ :
cd /usr/bin

跳到自己的 home directory :
cd ~

跳到目前目录的上上两层 :
cd ../..

指令名称 : chmod
使用权限 : 所有使用者

使用方式 : chmod [-cfvR] [--help] [--version] mode file...

说明 : Linux/Unix 的档案存取权限分为三级 : 档案拥有者、群组、其他。利用 chmod 可以藉以控制档案如何被他人所存取。

把计 :

mode : 权限设定字串，格式如下 : [ugoa...][[+-=][rwxX]...][,...]，其中u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。
-c : 若该档案权限确实已经更改，才显示其更改动作
-f : 若该档案权限无法被更改也不要显示错误讯息
-v : 显示权限变更的详细资料
-R : 对目前目录下的所有档案与子目录进行相同的权限变更(即以递回的方式逐个变更)
--help : 显示辅助说明
--version : 显示版本

范例 :将档案 file1.txt 设为所有人皆可读取 :
chmod ugo+r file1.txt

将档案 file1.txt 设为所有人皆可读取 :
chmod a+r file1.txt

将档案 file1.txt 与 file2.txt 设为该档案拥有者，与其所属同一个群体者可写入，但其他以外的人则不可写入 :
chmod ug+w,o-w file1.txt file2.txt

将 ex1.py 设定为只有该档案拥有者可以执行 :
chmod u+x ex1.py

将目前目录下的所有档案与子目录皆设为任何人可读取 :
chmod -R a+r *

此外chmod也可以用数字来表示权限如 chmod 777 file
语法为：chmod abc file

其中a,b,c各为一个数字，分别表示User、Group、及Other的权限。

r=4，w=2，x=1
若要rwx属性则4+2+1=7；
若要rw-属性则4+2=6；
若要r-x属性则4+1=7。

范例：
chmod a=rwx file

和
chmod 777 file

效果相同
chmod ug=rwx,o=x file

和
chmod 771 file

效果相同

若用chmod 4755 filename可使此程式具有root的权限

指令名称 : chown
使用权限 : root

使用方式 : chmod [-cfhvR] [--help] [--version] user[:group] file...

说明 : Linux/Unix 是多人多工作业系统，所有的档案皆有拥有者。利用 chown 可以将档案的拥有者加以改变。一般来说，这个指令只有是由系统管理者(root)所使用，一般使用者没有权限可以改变别人的档案拥有者，也没有权限可以自己的档案拥有者改设为别人。只有系统管理者(root)才有这样的权限。

把计 :

user : 新的档案拥有者的使用者 IDgroup : 新的档案拥有者的使用者群体(group)-c : 若该档案拥有者确实已经更改，才显示其更改动作-f : 若该档案拥有者无法被更改也不要显示错误讯息-h : 只对于连结(link)进行变更，而非该 link 真正指向的档案-v : 显示拥有者变更的详细资料-R : 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)--help : 显示辅助说明--version : 显示版本

范例 :
将档案 file1.txt 的拥有者设为 users 群体的使用者 jessie :
chown jessie:users file1.txt

将目前目录下的所有档案与子目录的拥有者皆设为 users 群体的使用者 lamport :
chmod -R lamport:users *

名称：cp
使用权限：所有使用者

使用方式：

cp [options] source dest
cp [options] source... directory

说明：将一个档案拷贝至另一档案，或将数个档案拷贝至另一目录。

把计

-a 尽可能将档案状态、权限等资料都照原状予以复制。
-r 若 source 中含有目录名，则将目录下之档案亦皆依序拷贝至目的地。
-f 若目的地已经有相同档名的档案存在，则在复制前先予以删除再行复制。
范例：
将档案 aaa 复制(已存在)，并命名为 bbb :
cp aaa bbb

将所有的C语言程式拷贝至 Finished 子目录中 :
cp *.c Finished

名称：cut

使用权限：所有使用者

用法：cut -cnum1-num2 filename

说明：显示每行从开头算起 num1 到 num2 的文字。

范例：

shell>> cat example
test2
this is test1
shell>> cut -c0-6 example ## print 开头算起前 6 个字元
test2
this i

名称 : find
用法 : find
使用说明 :

将档案系统内符合 expression 的档案列出来。你可以指要档案的名称、类别、时间、大小、权限等不

同资讯的组合，只有完全相符的才会被列出来。

find 根据下列规则判断 path 和 expression，在命令列上第一个 - ( ) , ! 之前的部份为 path，之后的是 expression。如果 path 是空字串则使用目前路径，如果 expression 是空字串则使用 -print 为预设 expression

expression 中可使用的选项有二三十个之多，在此只介绍最常用的部份。

-mount, -xdev : 只检查和指定目录在同一个档案系统下的档案，避免列出其它档案系统中的档案
-amin n : 在过去 n 分钟内被读取过
-anewer file : 比档案 file 更晚被读取过的档案
-atime n : 在过去 n 天过读取过的档案
-cmin n : 在过去 n 分钟内被修改过
-cnewer file :比档案 file 更新的档案
-ctime n : 在过去 n 天过修改过的档案
-empty : 空的档案-gid n or -group name : gid 是 n 或是 group 名称是 name
-ipath p, -path p : 路径名称符合 p 的档案，ipath 会忽略大小写
-name name, -iname name : 档案名称符合 name 的档案。iname 会忽略大小写
-size n : 档案大小 是 n 单位，b 代表 512 位元组的区块，c 表示字元数，k 表示 kilo bytes，w 是二个位元组。-type c : 档案类型是 c 的档案。
d: 目录
c: 字型装置档案
b: 区块装置档案
p: 具名贮列
f: 一般档案
l: 符号连结
s: socket
-pid n : process id 是 n 的档案

你可以使用 ( ) 将运算式分隔，并使用下列运算。
exp1 -and exp2
! expr
-not expr
exp1 -or exp2
exp1, exp2
范例:
将目前目录及其子目录下所有延伸档名是 c 的档案列出来。
find . -name "*.c"

将目前目录其其下子目录中所有一般档案列出
find . -ftype f

将目前目录及其子目录下所有最近 20 分钟内更新过的档案列出
find . -ctime -20

名称：less

使用权限：所有使用者

使用方式：

less [Option] filename

说明：
less 的作用与 more 十分相似，都可以用来浏览文字档案的内容，不同的是 less 允许使用者往回卷动

以浏览已经看过的部份，同时因为 less 并未在一开始就读入整个档案，因此在遇上大型档案的开启时，会比一般的文书编辑器(如 vi)来的快速。

范例：

指令名称 : ln
使用权限 : 所有使用者

使用方式 : ln [options] source dist，其中 option 的格式为 :

[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]
[--help] [--version] [--]
说明 : Linux/Unix 档案系统中，有所谓的连结(link)，我们可以将其视为档案的别名，而连结又可分为两种 : 硬连结(hard link)与软连结(symbolic link)，硬连结的意思是一个档案可以有多个名称，而软连结的方式则是产生一个特殊的档案，该档案的内容是指向另一个档案的位置。硬连结是存在同一个档案系统中，而软连结却可以跨越不同的档案系统。
ln source dist 是产生一个连结(dist)到 source，至于使用硬连结或软链结则由参数决定。

不论是硬连结或软链结都不会将原本的档案复制一份，只会占用非常少量的磁碟空间。

-f : 链结时先将与 dist 同档名的档案删除-d : 允许系统管理者硬链结自己的目录-i : 在删除与 dist 同档名的档案时先进行询问-n : 在进行软连结时，将 dist 视为一般的档案-s : 进行软链结(symbolic link)-v : 在连结之前显示其档名-b : 将在链结时会被覆写或删除的档案进行备份-S SUFFIX : 将备份的档案都加上 SUFFIX 的字尾-V METHOD : 指定备份的方式--help : 显示辅助说明--version : 显示版本
范例 :
将档案 yy 产生一个 symbolic link : zz
ln -s yy zz

将档案 yy 产生一个 hard link : zz
ln yy xx

名称：locate
使用权限：所有使用者
使用方式： locate [-q] [-d] [--database=]
locate [-r] [--regexp=]
locate [-qv] [-o] [--output=]
locate [-e] [-f] <[-l] [-c]
<[-U] [-u]>
locate [-Vh] [--version] [--help]
说明：
locate 让使用者可以很快速的搜寻档案系统内是否有指定的档案。其方法是先建立一个包括系统内所有档案名称及路径的资料库，之后当寻找时就只需查询这个资料库，而不必实际深入档案系统之中了。

在一般的 distribution 之中，资料库的建立都被放在 contab 中自动执行。一般使用者在使用时只要用

locate your_file_name

的型式就可以了。 参数：
-u
-U

建立资料库，-u 会由根目录开始，-U 则可以指定开始的位置。

-e

将
排除在寻找的范围之外。

-l
如果 是 1．则启动安全模式。在安全模式下，使用者不会看到权限无法看到的档案。这会始速度减慢，因为 locate 必须至实际的档案系统中取得档案的权限资料。

-f
将特定的档案系统排除在外，例如我们没有到理要把 proc 档案系统中的档案放在资料库中。

-q
安静模式，不会显示任何错误讯息。

-n
至多显示 个输出。

-r
使用正规运算式 做寻找的条件。

-o
指定资料库存的名称。

-d

指定资料库的路径

-h
显示辅助讯息

-v
显示更多的讯息

-V
显示程式的版本讯息 范例：

locate chdrv : 寻找所有叫 chdrv 的档案
locate -n 100 a.out : 寻找所有叫 a.out 的档案，但最多只显示 100 个
locate -u : 建立资料库

名称 : ls
使用权限 : 所有使用者

使用方式 : ls [-alrtAFR] [name...]

说明 : 显示指定工作目录下之内容（列出目前工作目录所含之档案及子目录)。

-a 显示所有档案及目录 (ls内定将档案名或目录名称开头为"."的视为隐藏档，不会列出)
-l 除档案名称外，亦将档案型态、权限、拥有者、档案大小等资讯详细列出
-r 将档案以相反次序显示(原定依英文字母次序)
-t 将档案依建立时间之先后次序列出
-A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)
-F 在列出的档案名称后加一符号；例如可执行档则加 "*", 目录则加 "/"
-R 若目录下有档案，则以下之档案亦皆依序列出

范例：
列出目前工作目录下所有名称是 s 开头的档案，愈新的排愈后面 :
ls -ltr s*

将 /bin 目录以下所有目录及档案详细资料列出 :
ls -lR /bin

列出目前工作目录下所有档案及目录；目录于名称后加 "/", 可执行档于名称后加 "*" :
ls -AF

名称：more
使用权限：所有使用者
使用方式：more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]
说明：类似 cat ，不过会以一页一页的显示方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h 。
参数：-num 一次显示的行数
-d 提示使用者，在画面下方显示 [Press space to continue, q to quit.] ，如果使用者按错键，则会显示 [Press h for instructions.] 而不是 哔 声
-l 取消遇见特殊字元 ^L（送纸字元）时会暂停的功能
-f 计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）
-p 不以卷动的方式显示每一页，而是先清除萤幕后再显示内容
-c 跟 -p 相似，不同的是先显示内容再清除其他旧资料
-s 当遇到有连续两行以上的空白行，就代换为一行的空白行
-u 不显示下引号 （根据环境变数 TERM 指定的 terminal 而有所不同）
+/ 在每个档案显示前搜寻该字串（pattern），然后从该字串之后开始显示
+num 从第 num 行开始显示
fileNames 欲显示内容的档案，可为复数个数
范例：
more -s testfile 逐页显示 testfile 之档案内容，如有连续两行以上空白行则以一行空白行显示。
more +20 testfile 从第 20 行开始显示 testfile 之档案内容。

名称：mv
使用权限：所有使用者

使用方式：

mv [options] source dest
mv [options] source... directory
说明：将一个档案移至另一档案，或将数个档案移至另一目录。
参数：-i 若目的地已有同名档案，则先询问是否覆盖旧档。

范例：

将档案 aaa 更名为 bbb :
mv aaa bbb

将所有的C语言程式移至 Finished 子目录中 :
mv -i *.c

名称：rm
使用权限：所有使用者

使用方式：rm [options] name...

说明：删除档案及目录。

把计

-i 删除前逐一询问确认。
-f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。
-r 将目录及以下之档案亦逐一删除。
范例：
删除所有C语言程式档；删除前逐一询问确认 :
rm -i *.c

将 Finished 子目录及子目录中所有档案删除 :
rm -r Finished

名称：rmdir
使用权限：于目前目录有适当权限的所有使用者

使用方式： rmdir [-p] dirName

说明： 删除空的目录。

参数： -p 是当子目录被删除后使它也成为空目录的话，则顺便一并删除。

范例：

将工作目录下，名为 AAA 的子目录删除 :
rmdir AAA

在工作目录下的 BBB 目录中，删除名为 Test 的子目录。若 Test 删除后，BBB 目录成为空目录，则 BBB 亦予删除。
rmdir -p BBB/Test

名称：split
使用权限：所有使用者

使用方式：split [OPTION] [INPUT [PREFIX]]

说明：

将一个档案分割成数个。而从 INPUT 分割输出成固定大小的档案，其档名依序为 PREFIXaa, PREFIXab...；PREFIX 预设值为 ****x。若没有 INPUT 档或为 ****-，则从标准输入读进资料。

匡兜

-b, --bytes=SIZE

SIZE 值为每一输出档案的大小，单位为 byte。
-C, --line-bytes=SIZE

每一输出档中，单行的最大 byte 数。
-l, --lines=NUMBER

NUMBER 值为每一输出档的列数大小。
-NUMBER

与 -l NUMBER 相同。
--verbose

于每个输出档被开启前，列印出侦错资讯到标准错误输出。
--help

显示辅助资讯然后离开。
--version

列出版本资讯然后离开。
SIZE 可加入单位: b 代表 512， k 代表 1K， m 代表 1 Meg。

范例：

PostgresSQL 大型资料库备份与回存：

因 Postgres 允许表格大过你系统档案的最大容量，所以要将表格 dump 到单一的档案可能会有问题，使用 split进行档案分割。

```
% pg_dump dbname | split -b 1m - filename.dump.
```

重新载入

```
% createdb dbname
% cat filename.dump.* | pgsql dbname
```

名称：touch

使用权限：所有使用者

使用方式：
touch [-acfm]
[-r reference-file] [--file=reference-file]
[-t MMDDhhmm[[CC]YY][.ss]]
[-d time] [--date=time] [--time={atime,access,use,mtime,modify}]
[--no-create] [--help] [--version]
file1 [file2 ...]

说明：
touch 指令改变档案的时间记录。 ls -l 可以显示档案的时间记录。

参数：
a 改变档案的读取时间记录。
m 改变档案的修改时间记录。
c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。
f 不使用，是为了与其他 unix 系统的相容性而保留。
r 使用参考档的时间记录，与 --file 的效果一样。
d 设定时间与日期，可以使用各种不同的格式。
t 设定档案的时间记录，格式与 date 指令相同。
--no-create 不会建立新档案。
--help 列出指令格式。
--version 列出版本讯息。

范例：

最简单的使用方式，将档案的时候记录改为现在的时间。若档案不存在，系统会建立一个新的档案。

touch file
touch file1 file2

将 file 的时间记录改为 5 月 6 日 18 点 3 分，公元两千年。时间的格式可以参考 date 指令，至少需输入 MMDDHHmm ，就是月日时与分。

touch -c -t 05061803 file
touch -c -t 050618032000 file

将 file 的时间记录改变成与 referencefile 一样。

touch -r referencefile file

将 file 的时间记录改成 5 月 6 日 18 点 3 分，公元两千年。时间可以使用 am, pm 或是 24 小时的

格式，日期可以使用其他格式如 6 May 2000 。

touch -d "6:03pm" file
touch -d "05/06/2000" file
touch -d "6:03pm 05/06/2000" file
名称 : at
使用权限 : 所有使用者

使用方式 : at -V [-q queue] [-f file] [-mldbv] TIME

说明 : at 可以让使用者指定在 TIME 这个特定时刻执行某个程式或指令，TIME 的格式是 HH:MM其中的 HH 为小时，MM 为分钟，甚至你也可以指定 am, pm, midnight, noon, teatime(就是下午 4 点锺)等口语词。

如果想要指定超过一天内的时间，则可以用 MMDDYY 或者 MM/DD/YY 的格式，其中 MM 是分钟，DD 是第几日，YY 是指年份。另外，使用者甚至也可以使用像是 now + 时间间隔来弹性指定时间，其中的时间间隔可以是 minutes, hours, days, weeks

另外，使用者也可指定 today 或 tomorrow 来表示今天或明天。当指定了时间并按下 enter 之后，at 会进入交谈模式并要求输入指令或程式，当你输入完后按下 ctrl+D 即可完成所有动作，至于执行的结果将会寄回你的帐号中。

把计 :

-V : 印出版本编号
-q : 使用指定的伫列(Queue)来储存，at 的资料是存放在所谓的 queue 中，使用者可以同时使用多个 queue，而 queue 的编号为 a, b, c... z 以及 A, B, ... Z 共 52 个
-m : 即使程式/指令执行完成后没有输出结果, 也要寄封信给使用者
-f file : 读入预先写好的命令档。使用者不一定要使用交谈模式来输入，可以先将所有的指定先写入档案后再一次读入
-l : 列出所有的指定 (使用者也可以直接使用 atq 而不用 at -l)
-d : 删除指定 (使用者也可以直接使用 atrm 而不用 at -d)
-v : 列出所有已经完成但尚未删除的指定

例子 :
三天后的下午 5 点锺执行 /bin/ls :
at 5pm + 3 days /bin/ls

三个星期后的下午 5 点锺执行 /bin/ls :
at 5pm + 2 weeks /bin/ls

明天的 17:20 执行 /bin/date :
at 17:20 tomorrow /bin/date

1999 年的最后一天的最后一分钟印出 the end of world !
at 23:59 12/31/1999 echo the end of world !

名称：cal

使用权限：所有使用者

使用方式：cal [-mjy] [month [year]]

说明：

显示日历。若只有一个参数，则代表年份(1-9999)，显示该年的年历。年份必须全部写出：********cal 89 将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。
1752 年 9 月第 3 日起改用西洋新历，因这时大部份的国家都采用新历，有 10 天被去除，所以该月份的月历有些不同。在此之前为西洋旧历。

匡兜

-m : 以星期一为每周的第一天方式显示。
-j : 以凯撒历显示，即以一月一日起的天数显示。
-y : 显示今年年历。

范例：

cal : 显示本月的月历。

```
[root@mylinux /root]# date
Tue Aug 15 08:00:18 CST 2000
[root@mylinux /root]# cal
August 2000
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[root@mylinux /root]#
```

cal 2001 : 显示公元 2001 年年历。

```
[root@mylinux /root]# cal 2001
2001
```

January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1 2 3
7 8 9 10 11 12 13 4 5 6 7 8 9 10 4 5 6 7 8 9 10
14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17
21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24
28 29 30 31 25 26 27 28 25 26 27 28 29 30 31

April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 5 1 2
8 9 10 11 12 13 14 6 7 8 9 10 11 12 3 4 5 6 7 8 9
15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16
22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23
29 30 27 28 29 30 31 24 25 26 27 28 29 30

July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 1 2 3 4 1
8 9 10 11 12 13 14 5 6 7 8 9 10 11 2 3 4 5 6 7 8
15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15
22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22
29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29
30
October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 1 2 3 1
7 8 9 10 11 12 13 4 5 6 7 8 9 10 2 3 4 5 6 7 8
14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15
21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22
28 29 30 31 25 26 27 28 29 30 23 24 25 26 27 28 29
30 31

[root@mylinux /root]#

cal 5 2001 : 显示公元 2001 年 5 月月历。

[root@mylinux /root]# cal 5 2001
May 2001
Su Mo Tu We Th Fr Sa

1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[root@mylinux /root]#

cal -m : 以星期一为每周的第一天方式，显示本月的月历。

[root@mylinux /root]# cal -m
August 2000
Mo Tu We Th Fr Sa Su
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

[root@mylinux /root]#

cal -jy : 以一月一日起的天数显示今年的年历。

[root@mylinux /root]# cal -jy
2000

January February
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
1 32 33 34 35 36
2 3 4 5 6 7 8 37 38 39 40 41 42 43
9 10 11 12 13 14 15 44 45 46 47 48 49 50
16 17 18 19 20 21 22 51 52 53 54 55 56 57
23 24 25 26 27 28 29 58 59 60
30 31
March April

Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
61 62 63 64 92
65 66 67 68 69 70 71 93 94 95 96 97 98 99
72 73 74 75 76 77 78 100 101 102 103 104 105 106
79 80 81 82 83 84 85 107 108 109 110 111 112 113
86 87 88 89 90 91 114 115 116 117 118 119 120
121
May June
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
122 123 124 125 126 127 153 154 155
128 129 130 131 132 133 134 156 157 158 159 160 161 162
135 136 137 138 139 140 141 163 164 165 166 167 168 169
142 143 144 145 146 147 148 170 171 172 173 174 175 176
149 150 151 152 177 178 179 180 181 182

July August
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
183 214 215 216 217 218
184 185 186 187 188 189 190 219 220 221 222 223 224 225
191 192 193 194 195 196 197 226 227 228 229 230 231 232
198 199 200 201 202 203 204 233 234 235 236 237 238 239
205 206 207 208 209 210 211 240 241 242 243 244
212 213
September October
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
245 246 275 276 277 278 279 280 281
247 248 249 250 251 252 253 282 283 284 285 286 287 288
254 255 256 257 258 259 260 289 290 291 292 293 294 295
261 262 263 264 265 266 267 296 297 298 299 300 301 302
268 269 270 271 272 273 274 303 304 305

November December
Sun Mon Tue Wed Thu Fri Sat Sun Mon Tue Wed Thu Fri Sat
306 307 308 309 336 337
310 311 312 313 314 315 316 338 339 340 341 342 343 344
317 318 319 320 321 322 323 345 346 347 348 349 350 351
324 325 326 327 328 329 330 352 353 354 355 356 357 358
331 332 333 334 335 359 360 361 362 363 364 365
366

[root@mylinux /root]#

名称 : crontab
使用权限 : 所有使用者

使用方式 :

crontab [-u user] filecrontab [-u user] { -l | -r | -e }
说明 :
crontab 是用来让使用者在固定时间或固定间隔执行程式之用，换句话说，也就是类似使用者的时程表。-u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

餐数 :

-e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说 setenv VISUAL joe)
-r : 删除目前的时程表
-l : 列出目前的时程表

时程表的格式如下 :
f1 f2 f3 f4 f5 program

其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程式。
当 f1 为 * 时表示每分钟都要执行 program，f2 为 * 时表示每小时都要执行程式，其余类推
当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推
当 f1 为 */n 时表示每 n 分钟个时间间隔执行一次，f2 为 */n 表示每 n 小时个时间间隔执行一次，其余类推
当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c... 个小时要执行，其余类推

使用者也可以将所有的设定先存放在档案 file 中，用 crontab file 的方式来设定时程表。
例子 :

每月每天每小时的第 0 分钟执行一次 /bin/ls :
0 7 * * * /bin/ls

在 12 月内, 每天的早上 6 点到 12 点中，每隔 20 分钟执行一次 /usr/bin/backup :
0 6-12/3 * 12 * /usr/bin/backup

周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata

每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分....执行 echo "haha"
20 0-23/2 * * * echo "haha"

注意 :

当程式在你所指定的时间执行后，系统会寄一封信给你，显示该程式执行的内容，若是你不希望收到这

样的信，请在每一行空一格之后加上 > /dev/null 2>&1 即可。

名称 : date
使用权限 : 所有使用者

使用方式 :

date [-u] [-d datestr] [-s datestr] [--utc] [--universal] [--date=datestr] [--set=datestr] [--help] [--version] [+FORMAT] [MMDDhhmm[[CC]YY][.ss]]

说明 :

date 可以用来显示或设定系统的日期与时间，在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中可用的标记列表如下 :

时间方面 :

% : 印出 %
%n : 下一行
%t : 跳格
%H : 小时(00..23)
%I : 小时(01..12)
%k : 小时(0..23)
%l : 小时(1..12)
%M : 分钟(00..59)
%p : 显示本地 AM 或 PM
%r : 直接显示时间 (12 小时制，格式为 hh:mmscreen.width-300)this.width=screen.width-300'>s [AP]M)
%s : 从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数
%S : 秒(00..61)
%T : 直接显示时间 (24 小时制)
%X : 相当于 %H:%M:%S
%Z : 显示时区

日期方面 :
%a : 星期几 (Sun..Sat)
%A : 星期几 (Sunday..Saturday)
%b : 月份 (Jan..Dec)
%B : 月份 (January..December)
%c : 直接显示日期与时间
%d : 日 (01..31)
%D : 直接显示日期 (mm/dd/yy)
%h : 同 %b
%j : 一年中的第几天 (001..366)
%m : 月份 (01..12)
%U : 一年中的第几周 (00..53) (以 Sunday 为一周的第一天的情形)
%w : 一周中的第几天 (0..6)

%W : 一年中的第几周 (00..53) (以 Monday 为一周的第一天的情形)
%x : 直接显示日期 (mm/dd/yy)
%y : 年份的最后两位数字 (00.99)
%Y : 完整年份 (0000..9999)

若是不以加号作为开头，则表示要设定时间，而时间格式为 MMDDhhmm[[CC]YY][.ss]，其中 MM 为月份，DD 为日，hh 为小时，mm 为分钟，CC 为年份前两位数字，YY 为年份后两位数字，ss 为秒数
把计 :

-d datestr : 显示 datestr 中所设定的时间 (非系统时间)
--help : 显示辅助讯息
-s datestr : 将系统时间设为 datestr 中所设定的时间
-u : 显示目前的格林威治时间
--version : 显示版本编号

例子 :
显示时间后跳行，再显示目前日期 :
date +%T%n%D

显示月份与日数 :
date +%B %d

显示日期与设定时间(12:34:56) :
date --date 12:34:56

注意 :

当你不希望出现无意义的 0 时(比如说 1999/03/07)，则可以在标记中插入 - 符号，比如说 date +%-H:%-M:%-S 会把时分秒中无意义的 0 给去掉，像是原本的 08:09:04 会变为 8:9:4。另外，只有取得权限者(比如说 root)才能设定系统时间。

当你以 root 身分更改了系统时间之后，请记得以 clock -w 来将系统时间写入 CMOS 中，这样下次重新开机时系统时间才会持续抱持最新的正确值。

名称 : sleep
使用权限 : 所有使用者

使用方式 : sleep [--help] [--version] number[smhd]

说明 : sleep 可以用来将目前动作延迟一段时间

参数说明 :

--help : 显示辅助讯息
--version : 显示版本编号
number : 时间长度，后面可接 s、m、h 或 d

其中 s 为秒，m 为 分钟，h 为小时，d 为日数

例子 :
显示目前时间后延迟 1 分钟，之后再次显示时间 :
date;sleep 1m;date

名称： time
使用权限： 所有使用者

使用方式： time [options] COMMAND [arguments]

说明： time 指令的用途，在于量测特定指令执行时所需消耗的时间及系统资源等资讯。例如 CPU 时间、记忆体、输入输出等等。需要特别注意的是，部分资讯在 Linux 上显示不出来。这是因为在 Linux 上部分资源的分配函式与 time 指令所预设的方式并不相同，以致于 time 指令无法取得这些资料。

把计

-o or --output=FILE
设定结果输出档。这个选项会将 time 的输出写入 所指定的档案中。如果档案已经存在，系统将覆写其内容。
-a or --append
配合 -o 使用，会将结果写到档案的末端，而不会覆盖掉原来的内容。
-f FORMAT or --format=FORMAT
以 FORMAT 字串设定显示方式。当这个选项没有被设定的时候，会用系统预设的格式。不过你可以用环境变数 time 来设定这个格式，如此一来就不必每次登入系统都要设定一次。
一般设定上，你可以用
t
表示跳栏，或者是用
n
表示换行。每一项资料要用 % 做为前导。如果要在字串中使用百分比符号，就用 。（学过 C 语言的人大概会觉得很熟悉）
time 指令可以显示的资源有四大项，分别是：

Time resources
Memory resources
IO resources
Command info

详细的内容如下：

Time Resources
E 执行指令所花费的时间，格式是：[hour]:minutescreen.width-300)this.width=screen.width-300'>econd。请注意这个数字并不代表实际的 CPU 时间。
e 执行指令所花费的时间，单位是秒。请注意这个数字并不代表实际的 CPU 时间。

S 指令执行时在核心模式（kernel mode）所花费的时间，单位是秒。
U 指令执行时在使用者模式（user mode）所花费的时间，单位是秒。
P 执行指令时 CPU 的占用比例。其实这个数字就是核心模式加上使用者模式的 CPU 时间除以总时间。

Memory Resources
M 执行时所占用的实体记忆体的最大值。单位是 KB
t 执行时所占用的实体记忆体的平均值，单位是 KB
K 执行程序所占用的记忆体总量（stack+data+text）的平均大小，单位是 KB
D 执行程序的自有资料区（unshared data area）的平均大小，单位是 KB
p 执行程序的自有堆叠（unshared stack）的平均大小，单位是 KB
X 执行程序间共享内容（shared text）的平均值，单位是 KB
Z 系统记忆体页的大小，单位是 byte。对同一个系统来说这是个常数

IO Resources
F 此程序的主要记忆体页错误发生次数。所谓的主要记忆体页错误是指某一记忆体页已经置换到置换档（swap file)中，而且已经分配给其他程序。此时该页的内容必须从置换档里再读出来。
R 此程序的次要记忆体页错误发生次数。所谓的次要记忆体页错误是指某一记忆体页虽然已经置换到置换档中，但尚未分配给其他程序。此时该页的内容并未被破坏，不必从置换档里读出来
W 此程序被交换到置换档的次数
c 此程序被强迫中断（像是分配到的 CPU 时间耗尽）的次数
w 此程序自愿中断（像是在等待某一个 I/O 执行完毕，像是磁碟读取等等）的次数
I 此程序所输入的档案数
O 此程序所输出的档案数
r 此程序所收到的 Socket Message
s 此程序所送出的 Socket Message
k 此程序所收到的信号 (Signal)数量

Command Info
C 执行时的参数以及指令名称
x 指令的结束代码 (Exit Status)

-p or --portability
这个选项会自动把显示格式设定成为：
real %e
user %U
sys %S
这么做的目的是为了与 POSIX 规格相容。
-v or --verbose
这个选项会把所有程式中用到的资源通通列出来，不但如一般英文语句，还有说明。对不想花时间去熟习格式设定或是刚刚开始接触这个指令的人相当有用。

范例：
利用下面的指令
time -v ps -aux

我们可以获得执行 ps -aux 的结果和所花费的系统资源。如下面所列的资料：
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 1096 472 ? S Apr19 0:04 init
root 2 0.0 0.0 0 0 ? SW Apr19 0:00 [kflushd]
root 3 0.0 0.0 0 0 ? SW Apr19 0:00 [kpiod]
......
root 24269 0.0 1.0 2692 996 pts/3 R 12:16 0:00 ps -aux

Command being timed: "ps -aux"
User time (seconds): 0.05
System time (seconds): 0.06
Percent of CPU this job got: 68%
Elapsed (wall clock) time (h:mmscreen.width-300)this.width=screen.width-300'>s or mscreen.width-300)this.width=screen.width-300'>s): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 0
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 238
Minor (reclaiming a frame) page faults: 46
Voluntary context switches: 0
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0

名称： uptime
使用权限： 所有使用者
使用方式： uptime [-V]
说明： uptime 提供使用者下面的资讯，不需其他参数：

现在的时间
系统开机运转到现在经过的时间
连线的使用者数量

最近一分钟，五分钟和十五分钟的系统负载
参数： -V 显示版本资讯。
范例： uptime
其结果为：
10:41am up 5 days, 10 min, 1 users, load average: 0.00, 0.00, 1.99

名称：chfn

使用权限：所有使用者

用法：shell>> chfn

说明：提供使用者更改个人资讯，用于 finger and mail username

范例：

shell>> chfn
Changing finger information for user
Password: [del]
Name[]:Johnney Huang ### 提供 finger 时的资料
Office[]:NCCU
Office Phone[]: [del]
Home Phone[]: [del]

名称：chsh

使用权限：所有使用者

用法：shell>> chsh

说明：更改使用者 shell 设定

范例：

shell>> chsh
Changing fihanging shell for user1
Password: [del]
New shell [/bin/tcsh]: ### [是目前使用的 shell]
[del]

shell>> chsh -l ### 展示 /etc/shells 档案内容
/bin/bash
/bin/sh
/bin/ash
/bin/bsh
/bin/tcsh

/bin/csh

” finger [返回]
名称： finger
使用权限： 所有使用者

使用方式： finger [options] user[@address]

说明：finger 可以让使用者查询一些其他使用者的资料。会列出来的资料有：

Login Name
User Name
Home directory
Shell
Login status
mail status
.plan
.project
.forward

其中 .plan ，.project 和 .forward 就是使用者在他的 Home Directory 里的 .plan ， .project 和 .forward 等档案里的资料。如果没有就没有。finger 指令并不限定于在同一伺服器上查询，也可以寻找某一个远端伺服器上的使用者。只要给一个像是 E-mail address 一般的地址即可。
把计

-l
多行显示。

-s
单行显示。这个选项只显示登入名称，真实姓名，终端机名称，闲置时间，登入时间，办公室号码及电话号码。如果所查询的使用者是远端伺服器的使用者，这个选项无效。

范例：下列指令可以查询本机管理员的资料：
finger root

其结果如下：
Login: root Name: root
Directory: /root Shell: /bin/bash
Never logged in.
No mail.
No Plan.

名称：last

使用权限：所有使用者

使用方式：shell>> last [options]

说明：显示系统开机以来获是从每月初登入者的讯息

把计

-R 省略 hostname 的栏位
-num 展示前 num 个
username 展示 username 的登入讯息
tty 限制登入讯息包含终端机代号

范例：

shell>> last -R -2
johnney pts/1 Mon Aug 14 20:42 still logged in
johnney pts/0 Mon Aug 14 19:59 still logged in

wtmp begins Tue Aug 1 09:01:10 2000 ### /var/log/wtmp

shell>> last -2 minery
minery pts/0 140.119.217.115 Mon Aug 14 18:37 - 18:40 (00:03)
minery pts/0 140.119.217.115 Mon Aug 14 17:22 - 17:24 (00:02)

wtmp begins Tue Aug 1 09:01:10 2000

名称:login

这个命令都不会就不要干算了！呵呵我也不在这里多费笔墨耽误大家美好青春了^_^

名称：passwd

使用权限：所有使用者

使用方式：passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]

说明：用来更改使用者的密码

参数：
-k

-l

-u

-f

-d 关闭使用者的密码认证功能, 使用者在登入时将可以不用输入密码, 只有具备 root 权限的使用者方可使用.

-S 显示指定使用者的密码认证种类, 只有具备 root 权限的使用者方可使用.

[username] 指定帐号名称.

名称 : who
使用权线 : 所有使用者都可使用

使用方式 : who - [husfV] [user]

说明 : 显示系统中有那些使用者正在上面，显示的资料包含了使用者 ID，使用的终端机，从那边连上来的，上线时间，呆滞时间，CPU 使用量，动作等等。

把计 :

-h : 不要显示标题列
-u : 不要显示使用者的动作/工作
-s : 使用简短的格式来显示
-f : 不要显示使用者的上线位置
-V : 显示程式版本
名称：/etc/aliases
使用权限：系统管理者
使用方式： 请用 newaliases 更新资料库
说明：
sendmail 会使用一个在 /etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据 aliases档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在 mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫 redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。

/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。newaliases 的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。范例：
newaliases

下面命令会做相同的事，
sendmail -bi

相关命令:
mail, mailq, newaliases, sendmail

” mail [返回]
名称：mail

使用权限：所有使用者

使用方式：mail [-iInv] [-s subject] [-c cc-addr] [-b bcc-addr] user1 [user 2 ...]

说明：
mail 不仅只是一个指令， mail 还是一个电子邮件程式，不过利用 mail 来读信的人应该很少吧！对于系统管理者来说 mail 就很有用，因为管理者可以用 mail 写成 script ，定期寄一些备忘录提醒系统的使用者。

参数：
i 忽略 tty 的中断讯号。 (interrupt)
I 强迫设成互动模式。 (Interactive)
v 列印出讯息，例如送信的地点、状态等等。 (verbose)
n 不读入 mail.rc 设定档。
s 邮件标题。
c cc 邮件地址。
b bcc 邮件地址。

范例：

将信件送给一个或以上的电子邮件地址，由于没有加入其他的选项，使用者必须输入标题与信件的内容等。而 user2 没有主机位置，就会送给邮件伺服器的 user2 使用者。

mail user1@email.address
mail user1@email.address user2

将 mail.txt 的内容寄给 user2 同时 cc 给 user1 。如果将这一行指令设成 cronjob 就可以定时将备忘录寄给系统使用者。

mail -s 标题 -c user1 user2 < mail.txt

指令：mesg
使用权限 : 所有使用者

使用方式 : mesg [y|n]

说明 ： 决定是否允许其他人传讯息到自己的终端机介面

把计

y : 允许讯息传到终端机介面上。
n : 不允许讯息传到终端机介面上 。
如果没有设定，则讯息传递与否则由终端机界面目前状态而定。

例子 :
改变目前讯息设定，改成不允许讯息传到终端机介面上 :
mesg n

与 mesg 相关的指令有： talk，write，wall。

名称：/etc/aliases
使用权限：系统管理者
使用方式： newaliases
说明：
sendmail 会使用一个在 /etc/aliases 中的档案做使用者名称转换的动作。当 sendmail 收到一个要送给 xxx 的信时，它会依据 aliases档的内容送给另一个使用者。这个功能可以创造一个只有在信件系统内才有效的使用者。例如 mailing list 就会用到这个功能，在 mailinglist 中，我们可能会创造一个叫 redlinux@link.ece.uci.edu 的 mailinglist，但实际上并没有一个叫 redlinux 的使用者。实际 aliases 档的内容是将送给这个使用者的信都收给 mailing list 处理程式负责分送的工作。

/etc/aliases 是一个文字模式的档案，sendmail 需要一个二进位格式的 /etc/aliases.db。
newaliases 的功能传是将 /etc/aliases 转换成一个 sendmail 所能了解的资料库。

参数：没有任何参数。 范例：
newaliases

下面命令会做相同的事，
sendmail -bi

相关命令:
mail, mailq, newaliases, sendmail

名称 : talk
使用权限 : 所有使用者

使用方式 :

talk person [ttyname]
说明 : 与其他使用者对谈

把计 :

person : 预备对谈的使用者帐号，如果该使用者在其他机器上，则可输入 person@machine.name
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

例子.1 :

与现在机器上的使用者Rollaend对谈，此时 Rollaend 只有一个连线 :
talk Rollaend

接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 ****talk jzlee****即可开始对谈，结束请按 ctrl+c

例子.2 :与linuxfab.cx上的使用者Rollaend对谈，使用pts/2来对谈 :
talk Rollaend@linuxfab.cx pts/2

接下来就是等Rollaend回应，若Rollaend接受，则Rollaend输入 ****talk jzlee@jzlee.home****即可开始对谈，结束请按 ctrl+c

注意 : 若萤幕的字会出现不正常的字元，试着按 ctrl+l 更新萤幕画面。

名称 : wall
使用权限 : 所有使用者

使用方式 :

wall [message]
使用说明：
wall 会将讯息传给每一个 mesg 设定为 yes 的上线使用者。当使用终端机介面做为标准传入时, 讯息结束时需加上 EOF (通常用 Ctrl+D)

例子 :

传讯息"hi" 给每一个使用者 :
wall hi

名称 : write
使用权限 : 所有使用者

使用方式 :

write user [ttyname]
说明 : 传讯息给其他使用者

把计 :

user : 预备传讯息的使用者帐号
ttyname : 如果使用者同时有两个以上的 tty 连线，可以自行选择合适的 tty 传讯息

例子.1 :

传讯息给 Rollaend，此时 Rollaend 只有一个连线 :
write Rollaend

接下来就是将讯息打上去，结束请按 ctrl+c

例子.2 :传讯息给 Rollaend，Rollaend 的连线有 pts/2，pts/3 :
write Rollaend pts/2

接下来就是将讯息打上去，结束请按 ctrl+c

注意 : 若对方设定 mesg n，则此时讯席将无法传给对方

名称：kill
使用权限：所有使用者

使用方式：

kill [-s signal | -p] [-a] pid ...
kill -l [signal]
说明：kill 送出一个特定的信号 (signal) 给行程 id 为 pid 的行程根据该信号而做特定的动作, 若没有指定, 预设是送出终止 (TERM) 的信号
把计

-s (signal) : 其中可用的讯号有 HUP (1), KILL (9), TERM (15), 分别代表着重跑, 砍掉, 结束; 详

细的信号可以用 kill -l
-p : 印出 pid , 并不送出信号
-l (signal) : 列出所有可用的信号名称
范例：

将 pid 为 323 的行程砍掉 (kill) :
kill -9 323

将 pid 为 456 的行程重跑 (restart) :
kill -HUP 456

名称：nice
使用权限：所有使用者

使用方式：nice [-n adjustment] [-adjustment] [--adjustment=adjustment] [--help] [--version] [command [arg...]]

说明：以更改过的优先序来执行程式, 如果未指定程式, 则会印出目前的排程优先序, 内定的 adjustment 为 10, 范围为 -20 (最高优先序) 到 19 (最低优先序)

把计

-n adjustment, -adjustment, --adjustment=adjustment 皆为将该原有优先序的增加 adjustment
--help 显示求助讯息
--version 显示版本资讯
范例：
将 ls 的优先序加 1 并执行 :
nice -n 1 ls

将 ls 的优先序加 10 并执行 :
nice ls将 ls 的优先序加 10 并执行

注意 : 优先序 (priority) 为作业系统用来决定 CPU 分配的参数，Linux 使用『回合制(round-robin)』的演算法来做 CPU 排程，优先序越高，所可能获得的 CPU时间就越多。

名称：ps
使用权限：所有使用者
使用方式：ps [options] [--help]
说明：显示瞬间行程 (process) 的动态
参数：
ps 的参数非常多, 在此仅列出几个常用的参数并大略介绍含义

-A 列出所有的行程
-w 显示加宽可以显示较多的资讯
-au 显示较详细的资讯
-aux 显示所有包含其他使用者的行程

au(x) 输出格式 :

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
USER: 行程拥有者
PID: pid
%CPU: 占用的 CPU 使用率
%MEM: 占用的记忆体使用率
VSZ: 占用的虚拟记忆体大小
RSS: 占用的记忆体大小
TTY: 终端的次要装置号码 (minor device number of tty)
STAT: 该行程的状态:
D: 不可中断的静止 (通悸□□缜b进行 I/O 动作)
R: 正在执行中
S: 静止状态
T: 暂停执行
Z: 不存在但暂时无法消除
W: 没有足够的记忆体分页可分配
<: 高优先序的行程
N: 低优先序的行程
L: 有记忆体分页分配并锁在记忆体内 (即时系统或捱A I/O)
START: 行程开始时间
TIME: 执行的时间
COMMAND:所执行的指令

范例:

ps
PID TTY TIME CMD
2791 ttyp0 00:00:00 tcsh
3092 ttyp0 00:00:00 ps
% ps -A
PID TTY TIME CMD
1 ? 00:00:03 init
2 ? 00:00:00 kflushd
3 ? 00:00:00 kpiod
4 ? 00:00:00 kswapd
5 ? 00:00:00 mdrecoveryd
....
% ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.7 1096 472 ? S Sep10 0:03 init [3]
root 2 0.0 0.0 0 0 ? SW Sep10 0:00 [kflushd]

root 3 0.0 0.0 0 0 ? SW Sep10 0:00 [kpiod]
root 4 0.0 0.0 0 0 ? SW Sep10 0:00 [kswapd]
名称：pstree
使用权限：所有使用者
使用方式：
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-G|-U] [pid|user]
pstree -V
说明：将所有行程以树状图显示，树状图将会以 pid (如果有指定) 或是以 init 这个基本行程为根 (root)，如果有指定使用者 id，则树状图会只显示该使用者所拥有的行程
参数：
-a 显示该行程的完整指令及参数，如果是被记忆体置换出去的行程则会加上括号
-c 如果有重覆的行程名，则分开列出 (预设值是会在前面加上 *
范例：

```
pstree

init-+-amd
|-apmd
|-atd
|-httpd---10*[httpd]
%pstree -p
init(1)-+-amd(447)
|-apmd(105)
|-atd(339)
%pstree -c
init-+-amd
|-apmd
|-atd
|-httpd-+-httpd
| |-httpd
| |-httpd
| |-httpd
....
```

名称：renice
使用权限：所有使用者

使用方式：renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]

说明：重新指定一个或多个行程(Process)的优先序(一个或多个将根据所下的参数而定)

把计

-p pid 重新指定行程的 id 为 pid 的行程的优先序
-g pgrp 重新指定行程群组(process group)的 id 为 pgrp 的行程 (一个或多个) 的优先序

-u user 重新指定行程拥有者为 user 的行程的优先序
范例：
将行程 id 为 987 及 32 的行程与行程拥有者为 daemon 及 root 的优先序号码加 1 :
renice +1 987 -u daemon root -p 32

注意 : 每一个行程(Process)都有一个唯一的 (unique) id

名称：top
使用权限：所有使用者

使用方式：top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]

说明：即时显示 process 的动态

把计

d : 改变显示的更新速度，或是在交谈式指令列(interactive command)按 s
q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称S : 累积模式，会将己完成或消失的子行程 (dead child process) 的 CPU time 累积起来
s : 安全模式，将交谈式指令取消, 避免潜在的危机
i : 不显示任何闲置 (idle) 或无用 (zombie) 的行程
n : 更新的次数，完成后将会退出 top
b : 批次档模式，搭配 "n" 参数一起使用，可以用来将 top 的结果输出到档案内

范例：
显示更新十次后退出 ;
top -n 10

使用者将不能利用交谈式指令来对行程下命令 :
top -s

将更新显示二次的结果输入到名称为 top.log 的档案里 :
top -n 2 -b < top.log

名称：skill

使用权限：所有使用者

使用方式： skill [signal to send] [options] 选择程序的规则

说明：

送个讯号给正在执行的程序,预设的讯息为 TERM (中断) , 较常使用的讯息为 HUP , INT , KILL , STOP , CONT ,和 0

讯息有三种写法:分别为 -9 , -SIGKILL , -KILL , 可以使用 -l 或 -L 已列出可使用的讯息。

一般参数：

-f 快速模式/尚未完成

-i 互动模式/ 每个动作将要被确认

-v 详细输出/ 列出所选择程序的资讯

-w 智能警告讯息/ 尚未完成

-n 没有动作/ 显示程序代号

参数：选择程序的规则可以是, 终端机代号,使用者名称,程序代号,命令名称。

-t 终端机代号 (tty 或 pty)

-u 使用者名称

-p 程序代号 (pid)

-c 命令名称 可使用的讯号:

以下列出已知的讯号名称,讯号代号,功能。

名称 (代号) 功能/ 描述

ALRM 14 离开

HUP 1 离开

INT 2 离开

KILL 9 离开/ 强迫关闭

PIPE 13 离开

POLL 离开

PROF 离开

TERM 15 离开

USR1 离开

USR2 离开

VTALRM 离开

STKFLT 离开/ 只适用于i386, m68k, arm 和 ppc 硬体

UNUSED 离开/ 只适用于i386, m68k, arm 和 ppc 硬体

TSTP 停止 /产生与内容相关的行为

TTIN 停止 /产生与内容相关的行为

TTOU 停止 /产生与内容相关的行为

STOP 停止 /强迫关闭

CONT 从新启动 /如果在停止状态则从新启动,否则忽略

PWR 忽略 /在某些系统中会离开

WINCH 忽略

CHLD 忽略

ABRT 6 核心

FPE 8 核心

ILL 4 核心

QUIT 3 核心

SEGV 11 核心

TRAP 5 核心

SYS 核心 /或许尚未实作

EMT 核心 /或许尚未实作

BUS 核心 /核心失败

XCPU 核心 /核心失败

XFSZ 核心 /核心失败

范例：

停止所有在 PTY 装置上的程序
skill -KILL -v pts/*

停止三个使用者 user1 , user2 , user3
skill -STOP user1 user2 user3

其他相关的命令: kill

名称：expr

使用权限：所有使用者
字串长度

shell>> expr length "this is a test"
14

数字商数

shell>> expr 14 % 9
5

从位置处抓取字串

shell>> expr substr "this is a test" 3 5
is is

数字串 only the first character

shell>> expr index "testforthegame" e
2

字串真实重现

shell>> expr quote thisisatestformela
thisisatestformela

名称: tr

1.比方说要把目录下所有的大写档名换为小写档名?

似乎有很多方式，"tr"是其中一种:

```
#!/bin/sh

dir="/tmp/testdir";
files=****find $dir -type f****;
for i in $files
do
dir_name=****dirname $i****;
ori_filename=****basename $i****
new_filename=****echo $ori_filename | tr [:upper:] [:lower:]**** > /dev/null;
#echo $new_filename;
mv $dir_name/$ori_filename $dir_name/$new_filename
done
```

2.自己试验中...lowercase to uppercase

```
tr abcdef...[del] ABCDE...[del]
tr a-z A-Z
tr [:lower:] [:upper:]

shell>> echo "this is a test" | tr a-z A-Z > www
shell>> cat www
THIS IS A TEST
```

3.去掉不想要的字串

```
shell>> tr -d this ### 去掉有关 t.e.s.t
this

man
man
test
e
```

4.取代字串

```
shell>> tr -s "this" "TEST"
this
TEST
th
TE
```

指令：clear
用途：清除萤幕用。

使用方法：在 console 上输入 clear。

名称: reset, tset
使用方法: tset [-IQqrs] [-] [-e ch] [-i ch] [-k ch] [-m mapping] [terminal]

使用说明:

reset 其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是 ? 的话，这个程式会要求使用者输入终端机的型别。

由于这个程式会将终端机设回原始的状态，除了在 login 时使用外，当系统终端机因为程式不正常执行而进入一些奇怪的状态时，你也可以用它来重设终端机o 例如不小心把二进位档用 cat 指令进到终端机，常会有终端机不再回应键盘输入，或是回应一些奇怪字元的问题。此时就可以用 reset 将终端机回复至原始状态。选项说明:

-p
将终端机类别显示在萤幕上，但不做设定的动作。这个命令可以用来取得目前终端机的类别。
-e ch
将 erase 字元设成 ch
-i ch
将中断字元设成 ch
-k ch
将删除一行的字元设成 ch
-I
不要做设定的动作，如果没有使用选项 -Q 的话，erase、中断及删除字元的目前值依然会送到萤幕上。

-Q
不要显示 erase、中断及删除字元的值到萤幕上。
-r
将终端机类别印在萤幕上。
-s
将设定 TERM 用的命令用字串的型式送到终端机中，通常在 .login 或 .profile 中用
范例:
让使用者输入一个终端机型别并将终端机设到该型别的预设状态。
reset ?

将 erase 字元设定 control-h
reset -e ^B

将设定用的字串显示在萤幕上
reset -s
Erase is control-B (^B).
Kill is control-U (^U).
Interrupt is control-C (^C).
TERM=xterm;

名称：compress

使用权限：所有使用者

使用方式：compress [-dfvcV] [-b maxbits] [file ...]

说明：
compress 是一个相当古老的 unix 档案压缩指令，压缩后的档案会加上一个 .Z 延伸档名以区别未压缩的档案，压缩后的档案可以以 uncompress 解压。若要将数个档案压成一个压缩档，必须先将档案 tar 起来再压缩。由于 gzip 可以产生更理想的压缩比例，一般人多已改用 gzip 为档案压缩工具。

参数：
c 输出结果至标准输出设备（一般指荧幕）
f 强迫写入档案，若目的档已经存在，则会被覆盖 (force)
v 将程式执行的讯息印在荧幕上 (verbose)
b 设定共同字串数的上限，以位元计算，可以设定的值为 9 至 16 bits 。由于值越大，能使用的共同字串就 越多，压缩比例就越大，所以一般使用预设值 16 bits (bits)
d 将压缩档解压缩
V 列出版本讯息

范例：

将 source.dat 压缩成 source.dat.Z ，若 source.dat.Z 已经存在，内容则会被压缩档覆盖。

compress -f source.dat

将 source.dat 压缩成 source.dat.Z ，并列印出压缩比例。
-v 与 -f 可以一起使用

compress -vf source.dat

将压缩后的资料输出后再导入 target.dat.Z 可以改变压缩档名。

compress -c source.dat > target.dat.Z

-b 的值越大，压缩比例就越大，范围是 9-16 ，预设值是 16 。

compress -b 12 source.dat

将 source.dat.Z 解压成 source.dat ，若档案已经存在，使用者按 y 以确定覆盖档案，若使用 -df 程式则会自动覆盖档案。由于系统会自动加入 .Z 为延伸档名，所以 source.dat 会自动当作 source.dat.Z 处理。

compress -d source.dat
compress -d source.dat.Z

名称： lpd

使用权限： 所有使用者

使用方式：lpd [-l] [#port]
lpd 是一个常驻的印表机管理程式，它会根据 /etc/printcap 的内容来管理本地或远端的印表机。/etc/printcap 中定义的每一个印表机必须在 /var/lpd 中有一个相对应的目录，目录中以 cf 开头的档案表示一个等待送到适当装置的印表工作。这个档案通常是由 lpr 所产生。

lpr 和 lpd 组成了一个可以离线工作的系统，当你使用 lpr 时，印表机不需要能立即可用，甚至不用存在。lpd 会自动监视印表机的状况，当印表机上线后，便立即将档案送交处理。这个得所有的应用程式不必等待印表机完成前一工作。

参数：

-l: 将一些除错讯息显示在标准输出上。
#port: 一般而言，lpd 会使用 getservbyname 取得适当的 TCP/IP port，你可以使用这个参数强迫 lpd 使用指定的 port。

范例：
这个程式通常是由 /etc/rc.d 中的程式在系统启始阶段执行。

名称 lpq
-- 显示列表机贮列中未完成的工作 用法

lpq [l] [P] [user]

说明
lpq 会显示由 lpd 所管理的列表机贮列中未完成的项目。

范例
范例 1. 显示所有在 lp 列表机贮列中的工作

lpq -PlpRank Owner Job Files Total Size1st root 238 (standard input) 1428646 bytes

相关函数
lpr,lpc,lpd

名称： lpr

使用权限： 所有使用者

使用方式：lpr [-P printer]
将档案或是由标准输入送进来的资料送到印表机贮列之中，印表机管理程式 lpd 会在稍后将这个档案送给适当的程式或装置处理。lpr 可以用来将料资送给本地或是远端的主机来处理。

参数：

-p Printer: 将资料送至指定的印表机 Printer，预设值为 lp。

范例：

将 http://www.c/和 kkk.c 送到印表机 lp。

lpr -Plp http://www.c/kkk.c

名称: lprm

-- 将一个工作由印表机贮列中移除 用法

/usr/bin/lprm [P] [file...]

说明

尚未完成的印表机工作会被放在印表机贮列之中，这个命令可用来将常未送到印表机的工作取消。由于每一个印表机都有一个独立的贮列，你可以用 -P 这个命令设定想要作用的印列机。如果没有设定的话，会使用系统预设的印表机。

这个命令会检查使用者是否有足够的权限删除指定的档案，一般而言，只有档案的拥有者或是系统管理员才有这个权限。

范例

将印表机 hpprinter 中的第 1123 号工作移除

lprm -Phpprinter 1123

将第 1011 号工作由预设印表机中移除

lprm 1011

名称： fdformat

使用权限： 所有使用者

使用方式：fdformat [-n] device

使用说明 :

对指定的软碟机装置进行低阶格式化。使用这个指令对软碟格式化的时候，最好指定像是下面的装置：

/dev/fd0d360 磁碟机 A: ，磁片为 360KB 磁碟

/dev/fd0h1440 磁碟机 A: ，磁片为 1.4MB 磁碟

/dev/fd1h1200 磁碟机 B: ，磁片为 1.2MB 磁碟

如果使用像是 /dev/fd0 之类的装置，如果里面的磁碟不是标准容量，格式化可能会失败。在这种情况之下，使用者可以用 setfdprm 指令先行指定必要参数。

参数:

-n 关闭确认功能。这个选项会关闭格式化之后的确认步骤。

范例：

fdformat -n /dev/fd0h1440

将磁碟机 A 的磁片格式化成 1.4MB 的磁片。并且省略确认的步骤。

名称： mformat

使用权限： 所有使用者

使用方式：

mformat [-t cylinders] [-h heads] [-s sectors] [-l volume_label] [-F] [-I fsVer-sion] [-S sizecode] [-2 sectors_on_track_0] [-M software_sector_size] [-a] [-X] [-C] [-H hidden_sectors] [-r root_sectors] [-B boot_sector] [-0 rate_on_track_0] [-A rate_on_other_tracks] [-1] [-k] drive:

在已经做过低阶格式化的磁片上建立 DOS 档案系统。如果在编译 mtools 的时候把 USE_2M 的参数打开，部分与 2M 格式相关的参数就会发生作用。否则这些参数（像是 S,2,1,M）不会发生作用。

参数：

-t 磁柱（synlider）数
-h 磁头（head）数
-s 每一磁轨的磁区数
-l 标签
-F 将磁碟格式化为 FAT32 格式，不过这个参数还在实验中。
-I 设定 FAT32 中的版本号。这当然也还在实验中。
-S 磁区大小代码，计算方式为 sector = 2^(大小代码+7)
-c 磁丛（cluster）的磁区数。如果所给定的数字会导致磁丛数超过 FAT 表的限制，mformat 会自动放大磁区数。
-s
-M 软体磁区大小。这个数字就是系统回报的磁区大小。通常是和实际的大小相同。
-a 如果加上这个参数，mformat 会产生一组 Atari 系统的序号给这块软碟。
-X 将软碟格式化成 XDF 格式。使用前必须先用 xdfcopy 指令对软碟作低阶格式化的动作。
-C 产生一个可以安装 MS-DOS 档案系统的磁碟影像档（disk image）。当然对一个实体磁碟机下这个参数是没有意义的。
-H 隐藏磁区的数目。这通常适用在格式化硬碟的分割区时，因为通常一个分割区的前面还有分割表。这个参数未经测试，能不用就不用。
-n 磁碟序号
-r 根目录的大小，单位是磁区数。这个参数只对 FAT12 和 FAT16 有效。
-B 使用所指定的档案或是设备的开机磁区做为这片磁片或分割区的开机磁区。当然当中的硬体参数会随

之更动。
-k 尽量保持原有的开机磁区。
-0 第 0 轨的资料传输率
-A 第 0 轨以外的资料传输率
-2 使用 2m 格式
-1 不使用 2m 格式

范例：
mformat a:

这样会用预设值把 a: （就是 /dev/fd0）里的磁碟片格式化。

名称： mkdosfs

使用权限： 所有使用者

使用方式： mkdosfs [-c | -l filename]
[-f number_of_FATs]
[-F FAT_size]
[-i volume_id]
[-m message_file]
[-n volume_name]
[-r root_dir_entry]
[-s sector_per_cluster]
[-v]
device
[block_count]

说明： 建立 DOS 档案系统。 device 指你想要建立 DOS 档案系统的装置代号。像是 /dev/hda1 等等。 block_count 则是你希望配置的区块数。如果 block_count 没有指定则系统会自动替你计算符合该装置大小的区块数。

参数：

-c 建立档案系统之前先检查是否有坏轨。
-l 从得定的档案中读取坏轨记录。
-f 指定档案配置表（FAT , File Allocation Table)的数量。预设值为 2 。目前 Linux 的 FAT 档案系统不支援超过 2 个 FAT 表。通常这个不需要改。
-F 指定 FAT 表的大小，通常是 12 或是 16 个位元组。12 位元组通常用于磁碟片，16 位元组用于一般硬碟的分割区，也就是所谓的 FAT16 格式。这个值通常系统会自己选定适当的值。在磁碟片上用 FAT16 通常不会发生作用，反之在硬碟上用 FAT12 亦然。
-i 指定 Volume ID。一般是一个 4 个位元组的数字，像是 2e203a47 。如果不给系统会自己产生。

-m 当使用者试图用这片磁片或是分割区开机，而上面没有作业系统时，系统会给使用者一段警告讯息。这个参数就是用来变更这个讯息的。你可以先用档案编辑好，然后用这个参数指定，或是用

-m -

这样系统会要求你直接输入这段文字。要特别注意的是，档案里的字串长度不要超过 418 个字，包括展开的跳栏符号（TAB）和换行符号（换行符号在 DOS 底下算两个字元！）

-n 指定 Volume Name，就是磁碟标签。如同在 DOS 底下的 format 指令一样，给不给都可以。没有预设值。

-r 指定根目录底下的最大档案数。这里所谓的档案数包括目录。预设值是在软碟上是 112 或是 224 ，在硬碟上是 512。没事不要改这个数字。

-s 每一个磁丛（cluster）的磁区数。必须是 2 的次方数。不过除非你知道你在作什么，这个值不要乱给。

-v 提供额外的讯息

范例：

mkdosfs -n Tester /dev/fd0 将 A 槽里的磁碟片格式化为 DOS 格式，并将标签设为 Tester

注：以上资料均来自于电子工业出版社《LINUX宝典大全》

上一页 目录 下一页

上一页 目录 下一页

Redhat9基于 NFS 的安装方法

在公司内部, 如果能进行Redhat 9基于 NFS 的安装将会给所有同事带来方便, 也不需要刻录光盘了, 而且安装也比较顺畅和简单。

一、下载rh9.0和检验MD5码:

1.下载：2003年4月11日更新

1. http://www.linuxiso.org/distro.php?distro=7

2. http://mirrors.kernel.org/redhat/redhat/linux/9/en/iso/（6CD)

3. http://download.enet.com.cn/search.php?condition=redhat%20linux&page=1&mode=1（据说50分钟可以下完）

2.检验：md5sum isofile > hashfile, 将 md5sum 档案与 hashfile 档案内容比对, 验证杂凑值是否一致 md5sum -c hashfile

二、安装系统

1. 配置 NFS 服务和拷贝文件：

（1）建立文件夹

mkdir /redhat9.0

cd /redhat9.0

（2）复制第一个镜象文件：mount $path-to/shrike-i386-disc1.iso /mnt/cdrom -o loop

cp /mnt/cdrom/Redhat . -r

umount
（3）复制第二个镜象文件:mount $path-to/shrike-i386-disc2.iso /mnt/cdrom -o loop
cp /mnt/cdrom/Redhat . -r
umount
（4）复制第三个镜象文件：mount $path-to/shrike-i386-disc2.iso /mnt/cdrom -o loop
cp /mnt/cdrom/Redhat . -r
umount

然后通过 redhat-config-nfs配置开放 /redhat9.0.

2. 制作安装启动光盘：

在 Redhat 9.0 的第一张盘上的 image 目录有一个 boot.iso 文件, 该文件是启动光盘的镜像文件, 将该文件刻录为启动光盘.

3. 通过 boot.iso 盘启动并安装系统, 选择 NFS 安装方式, NFS 路径为 /redhat9.0

上一页 目录 下一页

上一页 目录 下一页

REDHAT9上ADSL最终解决方案

REDHAT9上ADSL最终解决方案!

在LINUXSIR和LINUXFANS上看了很多关于ADSL的文章,都没有解决我的REDHAT9上ADSL上网的问题,今天实在是没有办法,重新建立连接,曲折的经历,终于上网了(非常激动,可能表达的不是很好),特的写下我的过程,作为参考:

REDHAT默认的PPPOE有问题,需要RPM -E,然后,安装这个
http://www.roaringpenguin.com/pppoe/rp-pppoe-3.5.tar.gz
解压和安装:
#tar zxvf rp-pppoe-3.5.tar.gz
进入解压目录执行
#sh ./go

然后再来设置ADSL。这一处，我们要用命令。

#adsl-setup

Welcome to the Roaring Penguin ADSL client setup. First, I will run
some checks on your system to make sure the PPPoE client is installed
properly...

Looks good! Now, please enter some information:

USER NAME

>>> Enter your PPPoE user name (default XXX): 在这里输入ADSL的用户名

INTERFACE

>>> Enter the Ethernet interface connected to the ADSL modem
For Solaris, this is likely to be something like /dev/hme0.
For Linux, it will be ethn, where 'n' is a number.
(default eth0):如果一张网卡就设置写上eth0

Do you want the link to come up on demand, or stay up continuously?
If you want it to come up on demand, enter the idle time in seconds
after which the link should be dropped. If you want the link to
stay up permanently, enter 'no' (two letters, lower-case.)
NOTE: Demand-activated links do not interact well with dynamic IP
addresses. You may have some problems with demand-activated links.
>>> Enter the demand value (default no):不用写什么

DNS
Please enter the IP address of your ISP's primary DNS server.
If your ISP claims that 'the server will provide DNS addresses',
enter 'server' (all lower-case) here.
If you just press enter, I will assume you know what you are
doing and not modify your DNS setup.
>>> Enter the DNS information here:在这里写上202.96.134.133
下一个DNS是202.96.168.68 //这里根据个人不同可以修改

PASSWORD

>>> Please enter your PPPoE password:输入密码
>>> Please re-enter your PPPoE password:再输入一次

FIREWALLING

Please choose the firewall rules to use. Note that these rules are
very basic. You are strongly encouraged to use a more sophisticated
firewall setup; however, these will provide basic security. If you
are running any servers on your machine, you must choose 'NONE' and
set up firewalling yourself. Otherwise, the firewall rules will deny
access to all standard servers like Web, e-mail, ftp, etc. If you
are using SSH, the rules will block outgoing SSH connections which
allocate a privileged source port.

The firewall choices are:
0 - NONE: This script will not set any firewall rules. You are responsible
for ensuring the security of your machine. You are STRONGLY
recommended to use some kind of firewall rules.
1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation
2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway
for a LAN
>>> Choose a type of firewall (0-2):这里添写为2

** Summary of what you entered **

Ethernet Interface: eth0
User name: XXX
Activate-on-demand: No
DNS: Do not adjust
Firewalling: MASQUERADE

>>> Accept these settings and adjust configuration files (y/n)?

弄完后，就按一个y键。
(以上为北南兄文章里面内容)
不要急于连接,REBOOT -N
然后进入网络设置,设置ETH1固定IP，停止ETH1(我的是用他)
然后ADSL-START
PING 你的DNS,如果可以,那么,恭喜你!
如果不行，REBOOT -N
ADSL-START,这下没问题啦，尽情享受吧!
其中部分内容可能不同,仅作参考,主要在连接后,能够PING通DNS即可!

上一页 目录 下一页

rpm命令参数列表

1.安装一个包
rpm -ivh
2.升级一个包
rpm -Uvh
3.移走一个包
rpm -e
4.安装参数
--force 即使覆盖属于其它包的文件也强迫安装

--nodeps 如果该RPM包的安装依赖其它包，即使其它包没装，也强迫安装。

5.查询一个包是否被安装

rpm -q < rpm package name>

6.得到被安装的包的信息

rpm -qi < rpm package name>

7.列出该包中有哪些文件

rpm -ql < rpm package name>

8.列出服务器上的一个文件属于哪一个RPM包

#rpm -qf

9.可综合好几个参数一起用

rpm -qil < rpm package name>

10.列出所有被安装的rpm package

rpm -qa

11.列出一个未被安装进系统的RPM包文件中包含有哪些文件？

rpm -qilp < rpm package name>

上一页 目录 下一页

上一页 目录 下一页

Vi编辑器的基本使用方法

vi编辑器是所有Unix及Linux系统下标准的编辑器，它的强大不逊色于任何最新的文本编辑器，这里只是简单地介绍一下它的用法和一小部分指令。由于对Unix及Linux系统的任何版本，vi编辑器是完全相同的，因此您可以在其他任何介绍vi的地方进一步了解它。Vi也是Linux中最基本的文本编辑器，学会它后，您将在Linux的世界里畅行无阻。

1、vi的基本概念

基本上vi可以分为三种状态，分别是命令模式（command mode）、插入模式（Insert mode）和底行模式（last line mode），各模式的功能区分如下：

1) 命令行模式command mode）

控制屏幕光标的移动，字符、字或行的删除，移动复制某区段及进入Insert mode下，或者到 last line mode。

2) 插入模式（Insert mode）

只有在Insert mode下，才可以做文字输入，按「ESC」键可回到命令行模式。

3) 底行模式（last line mode）

将文件保存或退出vi，也可以设置编辑环境，如寻找字符串、列出行号……等。

不过一般我们在使用时把vi简化成两个模式，就是将底行模式（last line mode）也算入命令行模式command mode）。

2、vi的基本操作
a) 进入vi

在系统提示符号输入vi及文件名称后，就进入vi全屏幕编辑画面：

$ vi myfile

不过有一点要特别注意，就是您进入vi之后，是处于「命令行模式（command mode）」，您要切换到「插入模式（Insert mode）」才能够输入文字。初次使用vi的人都会想先用上下左右键移动光标，结果电脑一直哔哔叫，把自己气个半死，所以进入vi后，先不要乱动，转换到「插入模式（Insert mode）」再说吧！

b) 切换至插入模式（Insert mode）编辑文件

在「命令行模式（command mode）」下按一下字母「i」就可以进入「插入模式（Insert mode）」，这时候你就可以开始输入文字了。

c) Insert 的切换

您目前处于「插入模式（Insert mode）」，您就只能一直输入文字，如果您发现输错了字！想用光标键往回移动，将该字删除，就要先按一下「ESC」键转到「命令行模式（command mode）」再删除文字。

d) 退出vi及保存文件

在「命令行模式（command mode）」下，按一下「：」冒号键进入「Last line mode」，例如：

w filename （输入 「w filename」将文章以指定的文件名filename保存）

wq (输入「wq」，存盘并退出vi)

q! (输入q!， 不存盘强制退出vi)

3、命令行模式（command mode）功能键
1）. 插入模式

按「i」切换进入插入模式「insert mode」，按“i”进入插入模式后是从光标当前位置开始输入文件；

按「a」进入插入模式后，是从目前光标所在位置的下一个位置开始输入文字；

按「o」进入插入模式后，是插入新的一行，从行首开始输入文字。

2）. 从插入模式切换为命令行模式

按「ESC」键。

3）. 移动光标

vi可以直接用键盘上的光标来上下左右移动，但正规的vi是用小写英文字母「h」、「j」、「k」、「l」，分别控制光标左、下、上、右移一格。

按「ctrl」+「b」：屏幕往“后”移动一页。

按「ctrl」+「f」：屏幕往“前”移动一页。

按「ctrl」+「u」：屏幕往“后”移动半页。

按「ctrl」+「d」：屏幕往“前”移动半页。

按数字「0」：移到文章的开头。

按「G」：移动到文章的最后。

按「$」：移动到光标所在行的“行尾”。

按「^」：移动到光标所在行的“行首”

按「w」：光标跳到下个字的开头

按「e」：光标跳到下个字的字尾

按「b」：光标回到上个字的开头

按「#l」：光标移到该行的第#个位置，如：5l,56l。

4）. 删除文字

「x」：每按一次，删除光标所在位置的“后面”一个字符。

「#x」：例如，「6x」表示删除光标所在位置的“后面”6个字符。

「X」：大写的X，每按一次，删除光标所在位置的“前面”一个字符。

「#X」：例如，「20X」表示删除光标所在位置的“前面”20个字符。

「dd」：删除光标所在行。

「#dd」：从光标所在行开始删除#行

5）. 复制

「yw」：将光标所在之处到字尾的字符复制到缓冲区中。

「#yw」：复制#个字到缓冲区

「yy」：复制光标所在行到缓冲区。

「#yy」：例如，「6yy」表示拷贝从光标所在的该行“往下数”6行文字。

「p」：将缓冲区内的字符贴到光标所在位置。注意：所有与“y”有关的复制命令都必须与“p”配合才能完成复制与粘贴功能。

6）. 替换

「r」：替换光标所在处的字符。

「R」：替换光标所到之处的字符，直到按下「ESC」键为止。

7）. 回复上一次操作

「u」：如果您误执行一个命令，可以马上按下「u」，回到上一个操作。按多次“u”可以执行多次回复。

8）. 更改

「cw」：更改光标所在处的字到字尾处

「c#w」：例如，「c3w」表示更改3个字

9）. 跳至指定的行

「ctrl」+「g」列出光标所在行的行号。

「#G」：例如，「15G」，表示移动光标至文章的第15行行首。

4、Last line mode下命令简介

在使用「last line mode」之前，请记住先按「ESC」键确定您已经处于「command mode」下后，再按「：」冒号即可进入「last line mode」。

A) 列出行号

「set nu」：输入「set nu」后，会在文件中的每一行前面列出行号。

B）跳到文件中的某一行

「#」：「#」号表示一个数字，在冒号后输入一个数字，再按回车键就会跳到该行了，如输入数字15，再回车，就会跳到文章的第15行。

C）查找字符

「关键字」：先按「」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往后寻找到您要的关键字为止。

「关键字」：先按「」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往前寻找到您要的关键字为止。

D）保存文件

「w」：在冒号输入字母「w」就可以将文件保存起来。

E）离开vi

「q」：按「q」就是退出，如果无法离开vi，可以在「q」后跟一个「!」强制离开vi。

「qw」：一般建议离开时，搭配「w」一起使用，这样在退出的时候还可以保存文件。

5、vi命令列表
1、下表列出命令模式下的一些键的功能：

h
左移光标一个字符

l
右移光标一个字符

k
光标上移一行

j
光标下移一行

^
光标移动至行首

0
数字“0”，光标移至文章的开头

G
光标移至文章的最后

$
光标移动至行尾

Ctrl+f
向前翻屏

Ctrl+b
向后翻屏

Ctrl+d
向前翻半屏

Ctrl+u
向后翻半屏

i
在光标位置前插入字符

a
在光标所在位置的后一个字符开始增加

o
插入新的一行，从行首开始输入

ESC
从输入状态退至命令状态

x
删除光标后面的字符

#x
删除光标后的＃个字符

X
(大写X)，删除光标前面的字符

#X
删除光标前面的#个字符

dd
删除光标所在的行

#dd
删除从光标所在行数的#行

yw
复制光标所在位置的一个字

#yw
复制光标所在位置的#个字

yy
复制光标所在位置的一行

#yy
复制从光标所在行数的#行

p
粘贴

u
取消操作

cw
更改光标所在位置的一个字

#cw
更改光标所在位置的#个字

2、下表列出行命令模式下的一些指令
w filename
储存正在编辑的文件为filename

wq filename
储存正在编辑的文件为filename，并退出vi

q!
放弃所有修改，退出vi

set nu
显示行号

或
查找，在后输入要查找的内容

n
与或一起使用，如果查找的内容不是想要找的关键字，按n或向后（与联用）或向前（与联用）继续查找，直到找到为止。

对于第一次用vi，有几点注意要提醒一下：
1、用vi打开文件后，是处于「命令行模式（command mode）」，您要切换到「插入模式（Insert mode）」才能够输入文字。切换方法：在「命令行模式（command mode）」下按一下字母「i」就可以进

入「插入模式（Insert mode）」，这时候你就可以开始输入文字了。
2、编辑好后，需从插入模式切换为命令行模式才能对文件进行保存，切换方法：按「ESC」键。
3、保存并退出文件：在命令模式下输入wq即可！（别忘了wq前面的）

上一页 目录 下一页

上一页 目录 下一页

VMware 安装RedHat 新手教程

(a).虚拟机简介：
虚拟机软件可以在一台电脑上模拟出来若干台PC，每台PC可以运行单独的操作系统而互不干扰，可以实现一台电脑“同时”运行几个操作系统。
目前PC上的虚拟机软件有下述两个：

VMWare http://www.vmware.com/
Virtual PC http://www.connectix.com/

我只介绍VMWare(主要是没用过Virtual PC)
可以在
http://www.piaodown.com/piao/downji/ji65.htm
下载注册机（ 本人不推荐）
注意：
虚拟机毕竟是将两台以上的电脑的任务集中在一台电脑上，所以对硬件的要求比较高，主要是CPU、硬盘和内存。所以如果机子比较烂，就不要装了！

VMWare用Host OS的文件来模拟Guest OS的硬盘。一个Guest OS的硬盘对应一个或多个Host OS里的文件。如果往Guest OS里写入100M的文件，Host OS里虚拟硬盘文件就增大100M。在Guest OS里删除这100M文件，Host OS里虚拟硬盘文件不会减小。

(b).拿到RedHat9.0安装光盘。没有光盘的话，在网上搞到iso映象也可以。
在虚拟机上安装系统和物理机上安装过程过程很类似。
下面开始：
运行VMWare Workstation 3.0
进入主界面，选择File->New Virtual Machine(直接点主界面右边那个图标也行）弹出一对话框，点Next,在Guest operating system下拉表里选择Linux,
再点Next,选择你想将虚拟机文件放置的地方。继续点Next,在这里选第二项NAT（我建议，你也可以选别的）这种方式也可以实现Host OS与Guest OS的双向访问。但网络内其他机器不能访问Guest OS，Guest OS可通过Host OS用NAT协议访问网络内其他机器。OK,点Finish!
现在在主界面Virtual Machine Name下选择我们刚刚建立的虚拟机，
打开Setting->Configuration Editor进行设置，我只讲一项：怎样增加一块虚拟硬盘，其他的自己去摸索 点底部的Add,蹦出一对话框，选Hard Disk,点Next,再Next,在这里调节虚拟硬盘大小，注意，但位是Gb,点Next,Click finish.OK!

开始真正的安装:
如果是物理光盘，请将第一张盘放入光驱。。。
if not ,打开Setting->Configuration Editor，选DVD/CD-ROM,
看对话框右边：选Use ISO image......我不想说了
好了，回到主界面，点Power On.。。。。当，当，当，当。。。
下面和真正装系统没多大区别，你看着办吧，哦，对了，
安装是一定要选择Text Mode ,
......我去喝会儿茶。。。
安装到配置Xserver时，选默认配置就可以，安装出错也不用理会。安装完后，先别起X。
现在你装好了吧，以root身份进入，我们要做一件惊天地，泣鬼神的大事，
安装vmware-tools，现在你已经在系统中了吧，点虚拟机菜单上的
Setting ->vmware-tools install,会弹出个对话框,按"确定"就行(选这个选项其实是虚拟一张光盘在/dev/cdrom里,这张虚拟光盘里有虚拟驱动),然后就:
mount /dev/cdrom /mnt/cdrom (将虚拟光盘挂上)
cd /mnt/cdrom
ls
应该会看见一个xxxvmtoolsxxx.tar.gz,将它拷贝到/tmp目录下,然后
tar zvxf xxx.tgz得到一个名字类似vmtools的目录
进入这个目录下,运行install.pl,一般来说,这个脚本会自动检测出你的linux版本,自动安装相应的驱动,一路确认就可以.
现在startx,是不是可以进入图形界面了。

安装GRUB

GRUB安装,配置及使用汇总- -

(一)首先要了解的几个概念

1.1 启动管理器
启动管理器是存储在磁盘开始扇区中的一段程序，例如，硬盘的MBR(Master Boot Record)，在系统完成启动测试后，如果系统是从MBR启动，则BIOS(Basic Input/Output System)将控制传送给MBR。然后存储在MBR中的这段程序将运行。这段程序被称为启动管理器。它的任务就是将控制传送给操作系统，完成启动过程。有许多可用的启动管理器，包括GNU GRUB (Grand Unified Boot Loader),Bootmanager, LILO (LInux LOader), NTLDR (boot loader for Windows NT systems)，等等等.
1.2 什么是GRUB?
grub 是一个多重启动管理器。grub是GRand Unified Bootloader的缩写，它可以在多个操作系统共存时选择引导哪个系统。它可以引导的操作系统包括：

Linux,FreeBSD,Solaris,NetBSD,BeOSi,OS/2,Windows95/98,Windows NT,Windows2000，WinXP。它可以载入操作系统的内核和初始化操作系统（如Linux,FreeBSD），或者把引导权交给操作系统（如Windows 98）来完成引导。

1.3 GRUB的特点

特别适用于linux与其它操作系统共存情况。

支持大硬盘 现在大多数Linux发行版本的lilo都有同样的一个问题：根分区(/boot分区)不能分在超过1024柱面的地方，一般是在8.4G左右的地方，否则lilo不能安装，或者安装后不能正确引导系统。而grub就不会出现这种情况，只要安装时你的大硬盘是在LBA模式下，grub就可以引导根分区在8G以外的操作系统。

支持开机画面　　grub支持在引导开机的同时显示一个开机画面。对于玩家来说，这样可以制作自己的个性化开机画面；对于PC厂商，这样可以在开机时显示电脑的一些信息和厂商的标志等。grub支持640x480,800x600,1024x768各种模式的开机画面，而且可以自动侦测选择最佳模式，与Windows那320x400的开机画面不可同日而语。

两种执行模式 grub不但可以通过配置文件进行例行的引导，还可以在选择引导前动态改变引导时的参数，还可以动态加载各种设备。例如你在Linux下编译了一个新的核心，但不能确定它能不能工作，你就可以在引导时动态改变grub的参数，尝试装载这个新的核心进行使用。Grub的命令行有非常强大的功能，而且支持如bash或doskey一样的历史功能，你可以用上下键来寻找以前的命令。

菜单式选择　　grub使用一个菜单来选择不同的系统进行引导。你还可以自己配置各种参数，如延迟时间，默认操作系统等。

分区大小改变后不必重新配置　　grub是通过文件系统直接把核心读取到内存，因此只要操作系统核心的路径没有改变，grub就可以引导系统。

除此之外，Grub还有许多非常强大的功能。例如支持多种外部设备，动态装载操作系统内核，甚至可以通过网络装载操作系统核心。Grub支持多种文件系统，支持多种可执行文件格式，支持自动解压，可以引导不支持多重引导的操作系统,支持网络启动等。

1.4 MBR和第一扇区

你可以简单的理解为MBR是整个硬盘的物理第一位置,而第一扇区是硬盘的物理第二位置.

(二)grub的配置

一旦选择了grub为引导程序,下面我们来了解一下它的配置.

/boot/grub/grub.conf是grub产生一个引导选择菜单以及设置一些选项.下面是我的grub.conf（menu.lst）

:

```
#==========例子开始==========
default=0
color light-gray/blue
timeout=10
splashimage=(hd0,6)/boot/grub/sw-splash.xpm.gz
title        Rays LX kernel 2.6.6-1-686-smp
root        (hd0,6)
kernel        /boot/vmlinuz-2.6.6-1-686-smp root=LABEL=/ ro vga=785 splash=silent
initrd        /boot/initrd.img-2.6.6-1-686-smp
savedefault
boot

title        Rays LX kernel 2.6.6-1-686-smp (recovery mode)
root        (hd0,6)
kernel        /boot/vmlinuz-2.6.6-1-686-smp root=LABEL=/ ro vga=785 splash=silent single
initrd        /boot/initrd.img-2.6.6-1-686-smp
```

```
savedefault
boot

title        Red Hat Linux 9
root        (hd0,7)
kernel        /vmlinuz-2.4.20-8 root=/dev/hda9 ro vga=785 splash=(hd0,7)/grub/splash.xpm.gz

initrd        /initrd-2.4.20-8.img
savedefault
boot
### END DEBIAN AUTOMAGIC KERNELS LIST

title    Windows 9x/2000/XP (on /dev/hda1)
root    (hd0,0)
makeactive
chainloader +1
######## Adding Windows bootable partitions End #########
#===========例子结束==========
```

配置选项解释:

以"#"开头的是注释行.

我这里有四个操作系统,分别是Rays LX、Red Hat Linux和Microsoft Windows XP.

Default：默认的操作系统就是由default控制的。default后加一个数字n，表明是第
n＋1个。如果你修改成3就是WinXP了

timeout：timeout标识默认等待时间,我这设置为10秒,超过10秒用户还没作出选择的话,将自动选择默认的操作系统。

Color：这个命令将生成彩色菜单。

splashimage一项指定grub界面的背景图片. 是开机画面文件sw-splash.xpm.gz的位置

title：显示选择菜单的名字。

root (hd0,6)标识从第一个硬盘,第7个分区来启动搜索引导内核.注意这儿的root与lin
ux的root分区不同,此root非彼root。grub的硬盘标识方法与linux的有点不同.在linux中第一个主分区为hda1,第二个主分区为hda2,第一个逻辑分区为hda5,而在grub中是以(hdx,y)来标识的,如第一个主分区为(hd0,0)第一个逻辑分区为(hd0,1)依此类推.所以这儿root后面的是你的/boot所在分区标识.

知道了内核在哪儿，还要具体指出哪个文件是内核文件，这就是kernel的工作。

kernel   /boot/vmlinuz-2.6.6-1-686-smp root=/dev/hda7 ro vga=785 splash=silent 和
kernel   /vmlinuz-2.4.20-8 root=/dev/hda9 ro vga=785 就是要载入的内核（第一个是Rays linux的内核,后面的是Red hat 9的）。后面的都是传递给内核的参数。ro是以readonly的意思，root=LABEL=/就是linux的硬盘分区表示法，vga=785是分辨率。注意我这里两个内核前面的路径。（Rays linux）是“/boot”,是因为我只为“Rays linux”挂载了根分区“/”而boot分区是在根分区之下的。而我为Red hat挂载了boot分区（hda8），所以这里直接就是“/”。

initrd用来初始的linux image，并设置相应的参数

再来看一看windows的定义段吧。

这里，我添加了一项来引导 WindowsXP。要完成此操作，GRUB 使用了"链式装入器"(ch
ainloader)，链式装入器从分区 (hd0,0)的引导记录中装入winXP自己的引导装入器NTLoader
，然后引导它。这就是这种技术叫做链式装入的原因 -- 它创建了一个从引导装入器到
另一个的链。这种链式装入技术可以用于引导任何版本的 DOS 或 Windows。如果你在计
算机中装有win98,winme,win2k,winxp的话,chainloader会把引导权交与win的NTLoader
来引导.

(三)Grub启动盘的制作
要制作引导盘，需执行一些简单的步骤。首先，在新的软盘上创建 ext2 文件系统。然后，将其安装，并将一些 GRUB 文件复制到该文件系统，最后运行 "grub" 程序，它将负责设置软盘的引导扇区。
将一张空盘插入 1.44MB 软驱，输入：
mke2fs /dev/fd0
创建了 ext2 文件系统后，需要安装该文件系统：
mount /dev/fd0 /mnt/floppy
现在，需要创建一些目录，并将一些关键文件（原先安装 GRUB 时已安装了这些文件）复制到软盘：
mkdir /mnt/floppy/boot
mkdir /mnt/floppy/boot/grub
cp /boot/grub/stage1 /mnt/floppy/boot/grub
cp /boot/grub/stage2 /mnt/floppy/boot/grub
再有一个步骤，就能得到可用的引导盘。
在linux bash中，从 root 用户运行"grub"，该程序非常有趣并值得注意，因为它实际上是 GRUB 引导装入器的半功能性版本。尽管 Linux 已经启动并正在运行，您仍可以运行 GRUB 并执行某些任务，而且其界面与使用 GRUB 引导盘或将 GRUB 安装到硬盘 MBR 时看到的界面(即GRUB控制台)完全相同。
在 grub> 提示符处，输入：
grub> root (fd0) （root 与（fd0）之间有空格，下同）
grub> setup (fd0)
grub> quit
现在，引导盘完成了。
(四).恢复被windows破坏的grub. （以Red hat 9为例）
如果你用grub来引导linux和windows,当windows出毛病重新安装后,会破坏MBR中的grub,这时需要恢复grub.（相当于用98启动软盘或光盘启动到实DOS下，用命令fdisk/mbr清除/mbr里grub的信息)
1.把linux安装光盘的第一张放到光驱，然后重新启动机器，在BOIS中把系统用光驱来引导。
2.等安装界面出来后键入linux rescue并按回车，进入linux rescue模式。
3.一系列键盘以及几项简单的配制，过后就［继续］了。。。这个过程，我不说了，比较简单。
4.然后会出现这样的提示符:sh# 。你可以到/mnt/下面看到一个sysimage这么目录，进去以后，就是你安装linux的/分区. 使用命令chroot将根分区变为当前目录的根分区:
chroot /mnt/sysimage
5.我们就可以操作GRUB了.输入grub:
sh#grub
会出现这样的提示符:
grub>
我们就可以在这样的字符后面，输入:
grub>root (hdX,Y)
grub>setup (hd0)
如果成功会有一个successful......
这里的X，如果是一个盘，就是0，如果你所安装的linux的根分区在第二个硬盘上，那X就是1了；Y，就是装有linux系统所在的根分区。 setup (hd0)就是把GRUB写到硬盘的MBR上。

(五).用NTLoader来引导linux.

如果你在安装linux时没有选择安装grub,不必着急,现在我们来看看如何在安装linux后安装grub.并用windows的NTLoader来引导linux.

1. 安装grub

我用的grub是Redhat带的grub安装包: grub-0.92-7.rpm

安装: rpm -ivh grub-0.92-7.rpm

其他安装方式也一样,只要你安装上grub就行了.（RH9缺省用的grub, 1,2步骤可以省了.）

2. 建立grub的环境

cp /usr/share/grub/i386-pc/* /boot/grub

3. 生成grub的配置文件/boot/grub/menu.conf

按照上面所讲的grub.conf来生成一个配置文件.

注意了, 这里我的linux在/dev/hda8,所以menu.conf那些分区位置为(hd0,7),

你的可能不一样了。下面第3步install的中的分区位置也应该和你的系统一致.

3. 安装grub至Linux分区boot

将grub的stage1安装到/dev/hda8的boot扇区(hd0,7). 过程如下:

/sbin/grub (运行grub)

grub> install (hd0,7)/grub/stage1 d (hd0,7) (hd0,7)/boot/grub/stage2 p

(hd0,7) /grub/menu.conf

(注意,上面"grub>"为grub的提示符.)

4. 取得grub的boot信息

过程如下:

dd if=/dev/hda8 of=/grub.lnx bs=512 count=1

if 是input_file of 是output_file bs是 字节数 count 是 执行的次数

这样在根目录下得到grub的引导信息,只要用NT Loader来加载它就行了.

5. 将上面得到的grub.lnx弄到Windows的C盘根目录下

可以先把grub.lnx弄得软盘上,然后启动windows,拷贝到C:\; 情况允许也可以直接在Linux下拷贝到C:了. 我的C盘(即设备/dev/hda1)为FAT32, 可以直接从Linux下弄过去了.

如下:

mount -t vfat /dev/hda1 /mnt/c

cp /grub.lnx /mnt/c

umount /mnt/c

6. 修改NT Loader的boot.ini

在其中加入一行: C:\grub.lnx="Redhat Linux - GRUB"

加入后boot.ini的内容如下:

[boot loader]

timeout=15

default=C:\boot.lnx

[operating systems]

multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /fastdetect

[VGA mode]" /basevideo /sos

C:\grub.lnx="Redhat Linux - GRUB"

OK. 可以用NT Loader加载Linux了, 其实上面过程基本思想就是用NT Loader来加载grub的引导区(grub.lnx), 其中的关键就是grub的引导区的获取. 。

(六)活用grub的交互功能

grub具有强大的交互功能.学会了将会使你受益非浅!

1.grub没有显示菜单怎么办?

当开机后进入grub界面但没了菜单,只剩下一个grub>提示符。

〈1〉 grub>cat (hd0,7)/grub/grub.conf(为了看参数，或者用fdisk -l显示当前分区情况)
〈2〉 grub>root (hd0,7)
〈3〉 grub>kernel (hd0,7)/boot/vmlinuz-2.6.6-1-686-smp root=LABEL=/ ro vga=785
〈4〉 grub>initrd (hd0,7)/initrd-2.4.18-14.img
〈5〉 grub>boot
以上这个方法也可以用于测试新编译的内核.
2.进入单用户模式.
有时不小心把root用户密码忘了,只能进入单用户模式来重新设置root密码。
进单用户模式的具体方法为：在出现grub引导界面的时候，按e，然后选中kernel/boot/vmlinuz-2.6.6-1-686-smp root=LABEL=/ ro vga=785 splash=silent这一项再按e编辑他，把他该成
kernel/boot/vmlinuz-2.6.6-1-686-smp root=LABEL=/ single ro vga=785 splash=silent之后按回车退出，再按b。
或者在在出现grub引导界面的时候然后按照上面的方法进行,只是在第三步要在后面加入single参数.
3. 设置GRUB的密码
〈1〉 进入grub字符cmd-line
＃grub
grub〉 md5crypt
Password: ********** 你的密码
Encrypted: 1U$JK7xFegdxWH6VuppCUSIb grub给出了密文
复制粘贴记下密文备用，如果忘记密文也无妨，重新md5crypt一下，因为同一个密码，grub给出的密文不一样
grub〉 reboot
＃

〈2〉 编辑menu.lst
＃vi /boot/grub/menu.lst
添加一行：password --md5 1U$JK7xFegdxWH6VuppCUSIb
password命令的位置可以不限，只要在所有title引导项之前就行了，然后是空格，两条短横线和md5，又是空格，你的密文。
　　现在要在启动时对GRUB菜单进行修改就不那么容易了，看菜单框下方的提示：你要敲“P”键，然后输入正确的密码才能对它动手动脚。
当然，这是不够的，它还能限制操作系统的访问。
在想要锁定的选项的title后一行加上lock命令：
title 在你想要的系统title项之后
lock 添加lock，即可实现进入本title系统，必须“p”之后才能进入的功能
root (hdx,x)

上一页 目录 下一页

常用配置

useradd admin

gpasswd -a admin root

modify /etc/passwd

find the last line where name is admin

modify 501 501 ===>>>> 0 0

/dev/hda5 /mnt/d vfat
defaults,codepage=936,iocharset=cp936
/dev/hda6 /mnt/e vfat
defaults,codepage=936,iocharset=cp936
/dev/hda7 /mnt/f vfat
defaults,codepage=936,iocharset=cp936
/dev/hda8 /mnt/g vfat
defaults,codepage=936,iocharset=cp936

如何让x0mms播放列表里显示正确的中文(myxfc)
-*-*-*-*-*-iso8859-1,-misc-simsun-medium-r-normal--12-*-*-*-*-*-gbk-0,*-r-

如何知道在redhat9下安装了哪些应用软件？

rpm -qa | grep filename

列出所有rpm软件。

rpm -qa

2.如何把一个目录下的所有文件权限改为777

chmod -R 777 ./*

3.如何知道自己的IP地址、路由网关、子网掩码？
[root@localhost root]# ifconfig ppp0

220.162.184.117就是的ip地址，220.162.184.1是路由网关，255.255.255.255是子网掩码

4.REDHAT可以自动获取ip地址和DNS服务器地址？
redhat-config-network

在里面将对应的网卡设成自动获取就可以了
5.哪个网站提供的rpm包最全最多？谢谢！

ftp://updates.redhat.com

6.能放mp3格式的播放器 xmms
(XMMS-1.2.10完美版)

http://www.linuxeden.com/download/

安装：tar jxvf xmms-1.2.10-wellexp.tar.bz

cd xmms-1.2.10-wellexp

./configure

make

make install

测试环境： redhatas3/redhat9

7.Xmms的mp3解压包
http://www.linuxeden.com/download/softdetail.php?softid=711的mp3解码包,直接安上就可以让rh8或rh9的xmms播放mp3了

install : rpm -ivh xmms-mp3-1.2.7-21.p.i386.rpm
[--nodeps --force]

如何让xmms播放列表里显示正确的中文(myxfc)
-*-*-*-*-*-iso8859-1,-misc-simsun-medium-r-normal--12-*-*-*-*-*-gbk-0,*-r-
把这个东西完全拷贝到你的字体里面
操作方法:
右键单击xmms播放工具的任何地方
会看到一个"选项",然后选择"功能设定"选择"fonts"
然后把上面的字体完整的拷贝到"播放清单"和 "user x font
（后者为可选项）

8.如何解压winrar和winzip这样的压缩包？

RAR for linux 3.30

下载地址：
http://www.linuxeden.com/download/softdetail.php?softid=1037
安装：tar zxvf rarlinux-3.3.0.tar.gz
cd rar
make
9.做ISO原来这么简单呵呵

cp /dev/cdrom /xxx.iso

我试过了，很好用
614M的用了不到5分钟就做好了
好使

10.如何让普通用户永远拥有超级用户的权限

修改/etc/passwd文件

#vi /etc/passwd
找到你要修改的用户名，
比如
hao:x:500:500::/home/hao:/bin/bash
把500改成0就可以了，保存退出。
第一个500是用户的uid(关键是改这个)，第二个500是用户的gid。
11.

播放影碟
从影碟中或者歌碟中拷贝avseq01.dat文件时，出现错误提示：无法读取/mnt/cdrom/mpegav/avseq01.dat。
但是读取其它的数据文件则没有任何问题。
我该该怎样播放影碟？

把.bat改为.asf就能播放了.

播放用mplayer,xine

拷贝用cp是不行的,得用抓轨工具 因为影碟的编码不是标准的iso9660

windos把这个弄的倒是挺简单的,蒙蔽了不少人

!
12.怎么在redhat下解压debian包？

1=>alien --to-rpm debianpackage.deb
将生成rpm包格式
2=>rpm -ivh 包
即可.
13.

如何用明令行方式调整音量？

man aumix

14.请问在redhat9.0下多个网卡如何设置IP
ifconfig ethX up

15.如何删除xwindow?

试试这个
rpm -qa|grep XFee86
rpm -e XFree86

16.请问mplayer怎么播放rm和rmvb?

将所有编解码器文件放入 /usr/local/lib/codecs/目录下即可。

17.普通用户不能访问/dev/dvd
chmod 666 /dev/dvd
一般来说，DVD-ROM驱动器在LINUX系统中被认为是在/dev/hdb1 或者 /dev/scd0，使用下面的命令即可：
ln -s /dev/cdrom /dev/dvd

18.NVIDIA Drivers 1.0-4363

http://download.nvidia.com/XFree86/Linux-x86/1.0-4363/NVIDIA_GLX-1.0-4363.tar.gz

19.请问如何取消MD5的验证?

rpm -ifvh filename.rpm

其中f参数是忽略任何错误的

20.关于安装NVIDIA显卡配置XF86Config文件的一些解答

,下面以RedHat 9.0 为例

.
下载：NVIDIA-Linux-x86-1.0-4349.run
http://www.linuxeden.com/download/softdetail.php?softid=832
或：http://download.nvidia.com/XFree86/Linux-x86/1.0-4363/NVIDIA_GLX-1.0-4363.tar.gz
1.备份XF86Config,即进入/etc/X11/目录，键入：
cp /etc/X11/XF86Config XF86Config.bk
2.进入下载文件的目录后，请在文本模式下键入：
sh ./NVIDIA-Linux-x86-1.0-4349.run
依据提示就可以顺利安装完毕。
3.用vi编辑器打开/etc/X11/XF86Config
配置文件，注释下列两：Load "dbe"和Load "dri"
即在它们的前面加上#号。
找到"section device"段内容，将其中的"device nv"
改成"device nvidia",保存，退出，进入图形界面就OK了。
21.挂载windows分区

在linux的分区表示中,硬盘为hd,第一块硬盘为hda,第二块为hdb.一块硬盘最多可以分成四个主分区,dos主分区,dos扩展分区,linux根分区和linux交换分区都属于主分区,4个主分区分别用数字表示,如果是第一块硬盘,就是hda1,hda2,hda3和hda4.在扩展分区上还可以分逻辑分区,标号从5往后依次排列.在windows中c盘为dos主分区,是hda1,d盘一般是dos扩展分区上的第一个逻辑分区,是hda5,e为hda6,f为hda7等等.在linux下可以通过mount命令挂栽windows分区到一个文件夹(这个文件夹称作挂载点),然后你可以通过这个文件夹访问windows分区,就象快捷方式.比如挂载c盘
先在/mnt/目录下新建目录winc
在终端输入mount -t vfat /dev/hda1 /mnt/winc -o codepage=936 iocharset=936(其中-t为指定设备文件系统,vfat表示windows的fat16和fat32文件系统格式,-o为指定加载文件系统时的选项,而选项codepage=936 iocharset=936是使linux可以正常显示挂载分区中的中文)
如果希望每次开机就自动加载windows分区,可以修改/etc/下的fstab文件,格式为/dev/hda1 /mnt/winc vfat defaults,codepage=936,iocharset=cp936 保存后重启即可.
22.支持ntfs分区
如果你的windows分区有ntfs格式的,要使linux支持ntfs格式,你可以通过编译内核实现,但最简单的方式是通过安装支持ntfs的rpm包实现.
请到 http://rpm.pbone.net/index.php3/stat/11/vendor/1620 下载相应内核的rpm(要想知道自己的内核版本请用如下命令查看：uname -a

)
我的rh9是kernel-ntfs-2.4.20-8.i686.rpm.然后单击右键,在快捷菜单里选择在此打开终端.在终端中输入 rpm -ivh kernel-ntfs-2.4.20-8.i686.rpm即可.(rpm为安装rpm包的命令,参数i 为显示软件包的相关信息,v 为显示命令执行过程,h为软件包安葬时列出标记)
22.安装Realplayer 9 rpm包
安装步骤:
(1)检查你是否按过openmotif21

rpm -q openmotif21

如果没有，就用天网搜一个装上，很好找得
我下载的是redhat9的，如下载openmotif21-2.1.30-8.i386.rpm，安装成功openmotif21-2.1.30-8.i386.rpm后就能找到那个什么libXm.so.2的文件了

(2)然后安装RealPlayer9-9.0.7.151-4.i386.rpm
3
http://cambuca.ldhs.cetuc.puc-rio.br/xine/
rpm -ivh RealPlayer9-9.0.7.151-4.i386.rpm

23.重装Win9x后lilo失效无法启动Linux
只须先用软盘启动Linux然后运行一次lilo即可。对于 RedHat可以用命令mkbootdisk来制作启动盘，例如： mkbootdisk --devices /dev/fd0 2.0.34-1 其中2.0.34-1是 /lib/modules下的一个目录。
[color=red:8ea1de57f3]那个2.0.34-1其实就是内核版本号,如果你不知道可以这样.mkbootdisk --devices /dev/fd0 `uname -r`.注意这个`是tab键上方的那个键,不是单引号.
[/color:8ea1de57f3]
24.Helix播放器(推荐)
HelixPlayer_linux-2.2-libc6-gcc32-i586.tar.gz
下载地址：
http://www.linuxeden.com/download/softdetail.php?softid=984
安装步骤：
tar -vxzf /path/to/HelixPlayer.tar.gz
例如：上述文件下载到了/root,请执行命令：
(tar -vxzf /root/HelixPlayer.tar.gz)
cd HelixPlayer
./hxplay
25. 安装xine
从 http://cambuca.ldhs.cetuc.puc-rio.br/xine/ 下载xine-ui-0.9.21cvs-030528.i586.rpm和libxine1-1_cvs-040410.i686.rpm(因为我是Pentium IV的CPU，也就是i686)
rpm -ivh libxine1-1_cvs-030528.k7.rpm
rpm -ivh xine-ui-0.9.21cvs-030528.i586.rpm
同样新建执行xine的应用程序链接

su - root
快速切换用户

自动mounut
在/etc/fstab 文件中加上 /dev/hda5 /mnt/d vfat devfaults iocharset=gb2312,umask=0 0 0

passwd -l 是索人帐号的啊!
解帐号,passwd -u好像哦!
自己也可以man passwd 看看!!
你可以先进入单用户模式啊!
首先打开电脑,在起动时,按E键,光标移到第二个再按E键.
然后 现在那个语句后面空格再打一个1(是数字一哦)或single俩选项一!
进入一个没密码的状态root状态!!
现在你就可以动行上面的命令了!!
不过不要黑人哦!!
你的进不了也许就是不小心用了passwd -l 用户名

这样就给停权了!!!

修改使命令行能够正常显示的配置文件是/etc/profile，具体操作如下
增加入下行到该配置文件中
stty cs8 -istrip
stty pass8
export LANG=C
export LC_CTYPE=iso-8859-1

1.首先下载Fedora Core 3 ISO (正在疯狂BT中....)复制到一独立的FAT32,或EXT2/3分区中. (hda6)
该分区不能在新系统安装中使用.
到
ftp.hostrino.com/pub/fedora/linux/core/2/i386/os/isolinux
下载引导程序.
copy 引导程序 to old linux's /boot/
2. vi /grub/grub.conf
add follow lines:
----
title Fedora Core 3 install
root (hd0,0)
kernel /vmlinuz
initrd /initrd.img

我的/boot是在独立的第一个分区,故root (hd0,0) path is "/"
reboot select Fedora Core 3 install
安装过程中选择硬盘安装,及ISO文件所在分区及目录.
3.往下安装过程同光盘安装.......

上一页 目录 下一页

上一页 目录 下一页

进程管理及作业控制

进程管理及作业控制(一)

Linux是一个多任务的操作系统，系统上同时运行着多个进程，正在执行的一个或多个相关进程称为一个作业。使用作业控制，用户可以同时运行多个作业，并在需要时在作业之间进行切换。本章详细介绍进

程管理及作业控制的命令，包括启动进程、查看进程、调度作业的命令。

进程及作业的概念

Linux是一个多用户多任务的操作系统。多用户是指多个用户可以在同一时间使用计算机系统；多任务是指Linux可以同时执行几个任务，它可以在还未执行完一个任务时又执行另一项任务。

操作系统管理多个用户的请求和多个任务。大多数系统都只有一个CPU和一个主存，但一个系统可能有多个二级存储磁盘和多个输入/输出设备。操作系统管理这些资源并在多个用户间共享资源，当您提出一个请求时，给您造成一种假象，好象系统只被您独自占用。而实际上操作系统监控着一个等待执行的任务队列，这些任务包括用户作业、操作系统任务、邮件和打印作业等。操作系统根据每个任务的优先级为每个任务分配合适的时间片，每个时间片大约都有零点几秒，虽然看起来很短，但实际上已经足够计算机完成成千上万的指令集。每个任务都会被系统运行一段时间，然后挂起，系统转而处理其他任务；过一段时间以后再回来处理这个任务，直到某个任务完成，从任务队列中去除。

Linux系统上所有运行的东西都可以称之为一个进程。每个用户任务、每个系统管理守护进程，都可以称之为进程。Linux用分时管理方法使所有的任务共同分享系统资源。我们讨论进程的时候，不会去关心这些进程究竟是如何分配的，或者是内核如何管理分配时间片的，我们所关心的是如何去控制这些进程，让它们能够很好地为用户服务。

进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。进程与程序是有区别的，进程不是程序，虽然它由程序产生。程序只是一个静态的指令集合，不占系统的运行资源；而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。而且一个程序可以启动多个进程。

Linux操作系统包括三种不同类型的进程，每种进程都有自己的特点和属性。

*

交互进程——由一个shell启动的进程。交互进程既可以在前台运行，也可以在后台运行。
*

批处理进程——这种进程和终端没有联系，是一个进程序列。
*

监控进程（也称守护进程）——Linux系统启动时启动的进程，并在后台运行。

上述三种进程各有各的作用，使用场合也有所不同。

进程和作业的概念也有区别。一个正在执行的进程称为一个作业，而且作业可以包含一个或多个进程，尤其是当使用了管道和重定向命令。例如“nroff -man ps.1|grep kill|more”这个作业就同时启动了三个进程。

作业控制指的是控制正在运行的进程的行为。比如，用户可以挂起一个进程，等一会儿再继续执行该进程。shell将记录所有启动的进程情况，在每个进程过程中，用户可以任意地挂起进程或重新启动进程。作业控制是许多shell（包括bash和tcsh）的一个特性，使用户能在多个独立作业间进行切换。

一般而言，进程与作业控制相关联时，才被称为作业。

在大多数情况下，用户在同一时间只运行一个作业，即它们最后向shell键入的命令。但是使用作业控制，用户可以同时运行多个作业，并在需要时在这些作业间进行切换。这会有什么用途呢？例如，当用户编辑一个文本文件，并需要中止编辑做其他事情时，利用作业控制，用户可以让编辑器暂时挂起，返回shell提示符开始做其他的事情。其他事情做完以后，用户可以重新启动挂起的编辑器，返回到刚才中止的地方，就象用户从来没有离开编辑器一样。这只是一个例子，作业控制还有许多其他实际的用途。

启动进程

键入需要运行的程序的程序名，执行一个程序，其实也就是启动了一个进程。在Linux系统中每个进程都具有一个进程号，用于系统识别和调度进程。启动一个进程有两个主要途径：手工启动和调度启动，后者是事先进行设置，根据用户要求自行启动。

手工启动

由用户输入命令，直接启动一个进程便是手工启动进程。但手工启动进程又可以分为很多种，根据启动的进程类型不同、性质不同，实际结果也不一样，下面分别介绍。

1. 前台启动

这或许是手工启动一个进程的最常用的方式。一般地，用户键入一个命令“ls －l”，这就已经启动了一个进程，而且是一个前台的进程。这时候系统其实已经处于一个多进程状态。或许有些用户会疑惑：我只启动了一个进程而已。但实际上有许多运行在后台的、系统启动时就已经自动启动的进程正在悄悄运行着。还有的用户在键入“ls －l”命令以后赶紧使用“ps －x”查看，却没有看到ls进程，也觉得很奇怪。其实这是因为ls这个进程结束太快，使用ps查看时该进程已经执行结束了。如果启动一个比较耗时的进程:

find / -name fox.jpg

然后再把该进程挂起，使用ps查看，就会看到一个find进程在里面。

2. 后台启动

直接从后台手工启动一个进程用得比较少一些，除非是该进程甚为耗时，且用户也不急着需要结果的时候。假设用户要启动一个需要长时间运行的格式化文本文件的进程。为了不使整个shell在格式化过程中都处于“瘫痪”状态，从后台启动这个进程是明智的选择。

［例1］

$ troff －me notes > note_form &

［1］ 4513

$

由上例可见，从后台启动进程其实就是在命令结尾加上一个&号。键入命令以后，出现一个数字，这个数字就是该进程的编号，也称为PID，然后就出现了提示符。用户可以继续其他工作。

上面介绍了前、后台启动的两种情况。实际上这两种启动方式有个共同的特点，就是新进程都是由当前shell这个进程产生的。也就是说，是shell创建了新进程，于是就称这种关系为进程间的父子关系。这里shell是父进程，而新进程是子进程。一个父进程可以有多个子进程，一般地，子进程结束后才能继续父进程；当然如果是从后台启动，那就不用等待子进程结束了。

一种比较特殊的情况是在使用管道符的时候。例如：

nroff -man ps.1|grep kill|more

这时候实际上是同时启动了三个进程。请注意是同时启动的，所有放在管道两边的进程都将被同时启动，它们都是当前shell的子程序，互相之间可以称为兄弟进程。

以上介绍的是手工启动进程的一些内容，作为一名系统管理员，很多时候都需要把事情安排好以后让其自动运行。因为管理员不是机器，也有离开的时候，所以有些必须要做的工作而恰好管理员不能亲自操作，这时候就需要使用调度启动进程了。

调度启动

有时候需要对系统进行一些比较费时而且占用资源的维护工作，这些工作适合在深夜进行，这时候用户就可以事先进行调度安排，指定任务运行的时间或者场合，到时候系统会自动完成这一切工作。

要使用自动启动进程的功能，就需要掌握以下几个启动命令。

at命令

用户使用at命令在指定时刻执行指定的命令序列。也就是说，该命令至少需要指定一个命令、一个执行时间才可以正常运行。at命令可以只指定时间，也可以时间和日期一起指定。需要注意的是，指定时间有个系统判别问题。比如说：用户现在指定了一个执行时间：凌晨3:20，而发出at命令的时间是头天晚上的 20:00，那么究竟是在哪一天执行该命令呢？如果用户在3:20以前仍然在工作，那么该命令将在这个时候完成；如果用户3:20以前就退出了工作状态，那么该命令将在第二天凌晨才得到执行。下面是at命令的语法格式：

at ［-V］ ［-q 队列］ ［-f 文件名］ ［-mldbv］ 时间

at -c 作业 ［作业...］

at允许使用一套相当复杂的指定时间的方法，实际上是将POSIX.2标准扩展了。它可以接受在当天的hh:mm（小时:分钟）式的时间指定。如果该时间已经过去，那么就放在第二天执行。当然也可以使用midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午4点）等比较模糊的词语来指定时间。用户还可以采用12小时计时制，即在时间后面加上AM（上午）或者PM（下午）来说明是上午还是下

午。

也可以指定命令执行的具体日期，指定格式为month day（月 日）或者mm/dd/yy（月/日/年）或者dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。

上面介绍的都是绝对计时法，其实还可以使用相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为：now + count time-units ，now就是当前时间，time-units是时间单位，这里可以是minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count是时间的数量，究竟是几天，还是几小时，等等。

还有一种计时方法就是直接使用today（今天）、tomorrow（明天）来指定完成命令的时间。下面通过一些例子来说明具体用法。

［例2］ 指定在今天下午5:30执行某命令。假设现在时间是中午12:30，1999年2月24日，其命令格式如下：

at 5:30pm

at 17:30

at 17:30 today

at now + 5 hours

at now + 300 minutes

at 17:30 24.2.99

at 17:30 2/24/99

at 17:30 Feb 24

以上这些命令表达的意义是完全一样的，所以在安排时间的时候完全可以根据个人喜好和具体情况自由选择。一般采用绝对时间的24小时计时法可以避免由于用户自己的疏忽造成计时错误的情况发生，例如上例可以写成：

at 17:30 2/24/99

这样非常清楚，而且别人也看得懂。

对于at命令来说，需要定时执行的命令是从标准输入或者使用-f选项指定的文件中读取并执行的。如果at命令是从一个使用su命令切换到用户shell中执行的，那么当前用户被认为是执行用户，所有的错误和输出结果都会送给这个用户。但是如果有邮件送出的话，收到邮件的将是原来的用户，也就是登录时shell的所有者。

［例3］

$ at -f work 4pm + 3 days

在三天后下午4点执行文件work中的作业。

$ at -f work 10am Jul 31

在7月31日上午10点执行文件work中的作业。

在任何情况下，超级用户都可以使用这个命令。对于其他用户来说，是否可以使用就取决于两个文件：/etc/at.allow和/etc/at.deny。如果/etc/at.allow文件存在的话，那么只有在其中列出的用户才可以使用at命令；如果该文件不存在，那么将检查/etc/at.deny文件是否存在，在这个文件中列出的用户均不能使用该命令。如果两个文件都不存在，那么只有超级用户可以使用该命令；空的/etc/at.deny文件意味着所有的用户都可以使用该命令，这也是默认状态。

下面对命令中的参数进行说明。

-V 将标准版本号打印到标准错误中。

-q queue 使用指定的队列。队列名称是由单个字母组成，合法的队列名可以由a-z或者A-Z。a队列是at命令的默认队列。

-m 作业结束后发送邮件给执行at命令的用户。

-f file 使用该选项将使命令从指定的file读取，而不是从标准输入读取。

-l atq命令的一个别名。该命令用于查看安排的作业序列，它将列出用户排在队列中的作业，如果是超级用户，则列出队列中的所有工作。

命令的语法格式如下:

atq ［-V］ ［-q 队列］ ［-v］

-d atrm 命令的一个别名。该命令用于删除指定要执行的命令序列，语法格式如下：

atrm ［-V］ 作业 ［作业...］

-c 将命令行上所列的作业送到标准输出。

［例4］ 找出系统中所有以txt为后缀名的文件，并且进行打印。打印结束后给用户foxy发出邮件通知取件。指定时间为十二月二十五日凌晨两点。

首先键入：

$ at 2:00 12/25/99

然后系统出现at>提示符，等待用户输入进一步的信息，也就是需要执行的命令序列：

at> find / -name “*.txt”|lpr

at> echo “foxy：All texts have been printed.You can take them over.Good day!River” |mail -s ”job done” foxy

输入完每一行指令然后回车，所有指令序列输入完毕后，使用<Ctrl+d>组合键结束at命令的输入。这时候屏幕将出现如下信息：

warning:command will be executed using /bin/sh.

job 1 at 1999-12-25 02:00

提醒用户将使用哪个shell来执行该命令序列。

实际上如果命令序列较长或者经常被执行的时候，一般都采用将该序列写到一个文件中，然后将文件作为at命令的输入来处理。这样不容易出错。

［例5］ 上面的例子可以修改如下：

将命令序列写入到文件/tmp/printjob，语句为：

$ at -f /tmp/printjob 2:00 12/25/99

这样一来，at命令将使用文件中的命令序列，屏幕显示如下：

Warning:command will be executed using /bin/sh.

job 2 at 1999-12-25 02:00

当然也可以采用以下命令：

$ at< /tmp/printjob 2:00 12/25/99

来完成同样的任务。也就是使用输入重定向的办法将文件定向为命令输入。

batch命令

batch 用低优先级运行作业，该命令几乎和at命令的功能完全相同，唯一的区别在于，at命令是在指定时间，很精确的时刻执行指定命令；而batch却是在系统负载较低，资源比较空闲的时候执行命令。该命令适合于执行占用资源较多的命令。

batch命令的语法格式也和at命令十分相似，即

batch [-V] [-q 队列] [-f 文件名] [-mv] [时间]

具体的参数解释请参考at命令。一般地说，不用为batch命令指定时间参数，因为batch本身的特点就是由系统决定执行任务的时间，如果用户再指定一个时间，就失去了本来的意义。

[例6] 使用例4，键入：

$ batch

at> find / -name *.txt|lpr

at> echo “foxy：All texts have been printed.You can take them over.Good day!River” |mail -s ”job done” foxy

现在这个命令就会在合适的时间进行了，进行完后会发回一个信息。

仍然使用<Ctrl+d>组合键来结束命令输入。而且batch和at命令都将自动转入后台，所以启动的时候也不需要加上&符号。

cron命令

前面介绍的两条命令都会在一定时间内完成一定任务，但是要注意它们都只能执行一次。也就是说，当指定了运行命令后，系统在指定时间完成任务，一切就结束了。但是在很多时候需要不断重复一些命令，比如：某公司每周一自动向员工报告头一周公司的活动情况，这时候就需要使用cron命令来完成任务了。

实际上，cron命令是不应该手工启动的。cron命令在系统启动时就由一个shell脚本自动启动，进入后台（所以不需要使用&符号）。一般的用户没有运行该命令的权限，虽然超级用户可以手工启动cron，不过还是建议将其放到shell脚本中由系统自行启动。

首先cron命令会搜索/var/spool/cron目录，寻找以/etc/passwd文件中的用户名命名的crontab文件，被找到的这种文件将载入内存。例如一个用户名为foxy的用户，它所对应的crontab文件就应该是/var/spool/cron/foxy。也就是说，以该用户命名的 crontab文件存放在/var/spool/cron目录下面。cron命令还将搜索/etc/crontab文件，这个文件是用不同的格式写成的。

cron启动以后，它将首先检查是否有用户设置了crontab文件，如果没有就转入“休眠”状态，释放系统资源。所以该后台进程占用资源极少。它每分钟 “醒”过来一次，查看当前是否有需要运行的命令。命令执行结束后，任何输出都将作为邮件发送给crontab的所有者，或者是/etc/crontab文件中MAILTO环境变量中指定的用户。

上面简单介绍了一些cron的工作原理，但是cron命令的执行不需要用户干涉；需要用户修改的是crontab中要执行的命令序列，所以下面介绍crontab命令。

crontab命令

crontab命令用于安装、删除或者列出用于驱动cron后台进程的表格。也就是说，用户把需要执行的命令序列放到crontab文件中以获得执行。每个用户都可以有自己的crontab文件。下面就来看看如何创建一

个crontab文件。

在/var/spool/cron下的crontab文件不可以直接创建或者直接修改。crontab文件是通过crontab命令得到的。现在假设有个用户名为foxy，需要创建自己的一个crontab文件。首先可以使用任何文本编辑器建立一个新文件，然后向其中写入需要运行的命令和要定期执行的时间。

然后存盘退出。假设该文件为/tmp/test.cron。再后就是使用crontab命令来安装这个文件，使之成为该用户的crontab文件。键入：

crontab test.cron

这样一个crontab 文件就建立好了。可以转到/var/spool/cron目录下面查看，发现多了一个foxy文件。这个文件就是所需的crontab 文件。用more命令查看该文件的内容可以发现文件头有三行信息：

#DO NOT EDIT THIS FILE -edit the master and reinstall.

#（test.cron installed on Mon Feb 22 14:20:20 1999）

#（cron version --$Id:crontab.c，v 2.13 1994/01/17 03:20:37 vivie Exp $）

大概意思是：

#切勿编辑此文件——如果需要改变请编辑源文件然后重新安装。

#test.cron文件安装时间：14:20:20 02/22/1999

如果需要改变其中的命令内容时，还是需要重新编辑原来的文件，然后再使用crontab命令安装。

可以使用crontab命令的用户是有限制的。如果/etc/cron.allow文件存在，那么只有其中列出的用户才能使用该命令；如果该文件不存在但 cron.deny文件存在，那么只有未列在该文件中的用户才能使用crontab命令；如果两个文件都不存在，那就取决于一些参数的设置，可能是只允许超级用户使用该命令，也可能是所有用户都可以使用该命令。

crontab命令的语法格式如下：

crontab ［-u user］ file

crontab ［-u user］{-l|-r|-e}

第一种格式用于安装一个新的crontab 文件，安装来源就是file所指的文件，如果使用“-”符号作为文件名，那就意味着使用标准输入作为安装来源。

-u 如果使用该选项，也就是指定了是哪个具体用户的crontab 文件将被修改。如果不指定该选项，crontab 将默认是操作者本人的crontab ，也就是执行该crontab 命令的用户的crontab 文件将被修改。但是请注意，如果使用了su命令再使用crontab 命令很可能就会出现混乱的情况。所以如果是使用

了su命令，最好使用-u选项来指定究竟是哪个用户的crontab文件。

-l 在标准输出上显示当前的crontab。

-r 删除当前的crontab文件。

-e 使用VISUAL或者EDITOR环境变量所指的编辑器编辑当前的crontab文件。当结束编辑离开时，编辑后的文件将自动安装。

［例7］

crontab -l #列出用户目前的crontab。

10 6 * * * date

0 */2 * * * date

0 23-7/2，8 * * * date

#

在crontab文件中如何输入需要执行的命令和时间。该文件中每行都包括六个域，其中前五个域是指定命令被执行的时间，最后一个域是要被执行的命令。每个域之间使用空格或者制表符分隔。格式如下：

minute hour day-of-month month-of-year day-of-week commands

第一项是分钟，第二项是小时，第三项是一个月的第几天，第四项是一年的第几个月，第五项是一周的星期几，第六项是要执行的命令。这些项都不能为空，必须填入。如果用户不需要指定其中的几项，那么可以使用*代替。因为*是统配符，可以代替任何字符，所以就可以认为是任何时间，也就是该项被忽略了。在表4-1 中给出了每项的合法范围。

表4-1　指定时间的合法范围

时间

合法值

minute

00-59

hour

00-23，其中00点就是晚上12点

day-of-month

01-31

month-of-year

01-12

day-of-week

0-6，其中周日是0

这样用户就可以往crontab 文件中写入无限多的行以完成无限多的命令。命令域中可以写入所有可以在命令行写入的命令和符号，其他所有时间域都支持列举，也就是域中可以写入很多的时间值，只要满足这些时间值中的任何一个都执行命令，每两个时间值中间使用逗号分隔。

［例8］

5，15，25，35，45，55 16，17，18 * * * command

这就是表示任意天任意月，其实就是每天的下午4点、5点、6点的5 min、15 min、25 min、35 min、45 min、55 min时执行命令。

［例9］ 在每周一，三，五的下午3：00系统进入维护状态，重新启动系统。那么在crontab 文件中就应该写入如下字段：

00 15 * * 1，3，5 shutdown -r +5

然后将该文件存盘为foxy.cron，再键入crontab foxy.cron安装该文件。

［例10］ 每小时的10分，40分执行用户目录下的innd/bbslin这个指令：

10，40 * * * * innd/bbslink

［例11］ 每小时的1分执行用户目录下的bin/account这个指令：

1 * * * * bin/account

［例12］ 每天早晨三点二十分执行用户目录下如下所示的两个指令（每个指令以;分隔）：

20 3 * * * （/bin/rm -f expire.ls logins.bad;bin/expire>expire.1st）

［例13］ 每年的一月和四月，4号到9号的3点12分和3点55分执行/bin/rm -f expire.1st这个指令，并把结果添加在mm.txt这个文件之后（mm.txt文件位于用户自己的目录位置）。

12,55 3 4-9 1,4 * /bin/rm -f expire.1st>>mm.txt

［例14］ 我们来看一个超级用户的crontab文件：

#Run the ‘atrun’ program every minutes

#This runs anything that’s due to run from ‘at’.See man ‘at’ or ‘atrun’.

0,5,10,15,20,25,30,35,40,45,50,55 * * * * /usr/lib/atrun

40 7 * * * updatedb

8,10,22,30,39,46,54,58 * * * * /bin/sync

进程的挂起及恢复命令bg、fg

作业控制允许将进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按<ctrl+z>，即可挂起当前的前台作业。

［例15］

$ cat > text.file

<ctrl+z>

［1］ + stopped cat > text.file

$ jobs

［1］+ stopped cat > text.file

在键盘上按<ctrl+z>后，将挂起当前执行的命令cat。使用jobs命令可以显示shell的作业清单，包括具体的作业、作业号以及作业当前所处的状态。

恢复进程执行时，有两种选择：用fg命令将挂起的作业放回到前台执行；用bg命令将挂起的作业放到后台执行。

［例16］用户正在使用Emacs，突然需要查看系统进程情况。就首先使用<Ctrl+z>组合键将Emacs进程挂起，然后使用bg命令将其在后台启动，这样就得到了前台的操作控制权，接着键入“ps －x”查看进程情况。查看完毕后，使用fg命令将Emacs带回前台运行即可。其命令格式为：

<Ctrl+z>

$ bg emacs

$ ps －x

```
$ fg emacs
```

默认情况下，fg和bg命令对最近停止的作业进行操作。如果希望恢复其他作业的运行，可以在命令中指定要恢复作业的作业号来恢复该作业。例如：

```
$ fg 1

cat > text.file
```

灵活使用上述命令，将给自己带来很大的方便。

上一页 目录 下一页

上一页 目录 下一页

命令大集合

进入与退出系统

TurboLinux是一个多用户的操作系统，用户要使用该系统，首先必须登录系统，使用完系统后，必须退出系统。本章主要讨论登录和退出系统的方法。

用户登录系统时，为了使系统能够识别自己，必须输入用户名和密码，经系统验证无误后方能进入系统。在系统安装过程中可以创建两种帐号：

1）root--超级用户帐号，使用这个帐号可以在系统中做任何事情。

2）普通用户--这个帐号供普通用户使用，可以进行有限的操作。

一般的Linux使用者均为普通用户，而系统管理员一般使用超级用户帐号完成一些系统管理的工作。如果只需要完成一些由普通帐号就能完成的任务，建议不要使用超级用户帐号，以免无意中破坏系统。

用户登录分两步进行：第一步，输入用户的登录名，系统根据该登录名来识别用户；第二步，输入用户的口令，该口令是用户自己选择的一个字符串，对其他用户是保密的，是在登录时系统用来辨别真假用户的关键字。

在Linux系统中，系统管理员在为用户建立新帐号时赋给用户一个用户名和一个初始的口令。另外，Linux系统给计算机赋予一个主机名。主机名用于在网络上识别独立的计算机（即使用户的计算机没有联网，也应该有一个主机名）。TurboLinux系统给出的缺省主机名为：localhost。在下面的例子中，我们假设用户名为"xxq"，系统的主机名为"localhost"。

进入系统（登录）

超级用户登录

超级用户的用户名为root，密码在安装系统时已设定。系统启动成功后，屏幕显示下面的提示：

localhost login:

这时输入超级用户名"root"，然后键入回车键。此时，用户会在屏幕上看到输入口令的提示：

localhost login:root

Password:

这时，需要输入口令。输入口令时，口令不会在屏幕上显示出来。如果用户输入了错误的口令，就会在屏幕上看到下列信息：

login incorrect.

这时需要重新输入。当用户正确地输入用户名和口令后，就能合法地进入系统。屏幕显示：

［root@loclhost /root］ #

此时说明该用户已经登录到系统中，可以进行操作了。这里"#"是超级用户的系统提示符。普通用户登录建立了普通用户帐号以后，就可以进行登录了。

在登录时，用户会在屏幕上看到类似下面的提示：

localhost login:

这时输入用户名"xxq"，然后键入回车键。此时，用户会在屏幕上看到输入口令的提示：

localhost login:xxq

Password:

这时，需要输入口令。输入口令时，口令不会在屏幕上显示出来。如果用户输入了错误的口令，就会在屏幕上看到下列信息：

login incorrect.

这时需要重新输入。当用户正确地输入用户名和口令后，就能合法地进入系统。屏幕显示：

［xxq@loclhost xxq］ $

此时说明该用户已经登录到系统中，可以进行操作了。

修改口令

为了更好的保护用户帐号的安全，Linux允许用户随时修改自己的口令，修改口令的命令是passwd，它将提示用户输入旧口令和新口令，之后还要求用户再次确认新口令，以避免用户无意中按错键。如果用户忘记了口令，可以请系统管理员为自己重新设置一个。

虚拟控制台

Linux是一个真正的多用户操作系统，这表示它可以同时接受多个用户登录。Linux还允许一个用户进行多次登录，这是因为Linux和许多版本的UNIX一样，提供了虚拟控制台的访问方式，允许用户在同一时间从控制台（系统的控制台是与系统直接相连的监视器和键盘）进行多次登录。

虚拟控制台的选择可以通过按下Alt键和一个功能键来实现，通常使用F1-F6。

例如，用户登录后，按一下Alt-F2键，用户又可以看到"login:"提示符，说明用户看到了第二个虚拟控制台。然后只需按Alt-F1键，就可以回到第一个虚拟控制台。一个新安装的Linux系统允许用户使用Alt-F1到Alt-F6键来访问前六个虚拟控制台。

虚拟控制台可使用户同时在多个控制台上工作，真正感受到Linux系统多用户的特性。用户可以在某一虚拟控制台上进行的工作尚未结束时，切换到另一虚拟控制台开始另一项工作。例如，开发软件时，可以在一个控制台上进行编辑，在另一个控制台上进行编译，在第三个控制台上查阅信息。

退出系统

不论是超级用户，还是普通用户，需要退出系统时，在shell提示符下，键入下列命令即可。

下面以普通用户的退出为例，说明退出系统的过程：

［xxq@loclhost xxq］ $ exit

还有其他退出系统的方法，但上面一种是最安全的。

屏幕文本编辑器Vi

本章介绍Linux上最常用的文本编辑器Vi。文本编辑器是所有计算机系统中最常使用的一种工具。用户在使用计算机的时候，往往需要建立自己的文件，无论是一般的文本文件、数据文件，还是编写的源程序文件，这些工作都离不开编辑器。

Linux系统提供了一个完整的编辑器家族系列，如Ed、Ex、Vi和Emacs等，按功能它们可以分为两大类：行编辑器（Ed、Ex）和全屏幕编辑器（Vi、Emacs）。行编辑器每次只能对一行进行操作，使用起来很不方便。而全屏幕编辑器可以对整个屏幕进行编辑，用户编辑的文件直接显示在屏幕上，修改的结果可以立即看出来，克服了行编辑的那种不直观的操作方式，便于用户学习和使用，具有强大的功能。

Vi是Linux系统的第一个全屏幕交互式编辑程序，它从诞生至今一直得到广大用户的青睐，历经数十年仍然是人们主要使用的文本编辑工具，足见其生命力之强，而强大的生命力是其强大的功能带来的。

本章中，我们将循序渐进地介绍如何使用Vi来建立、编辑、显示以及处理文件。（注：文中的大写字母为［shift］键+相应的小写字母的组合）

Vi简介

Vi是“Visual interface”的简称，它在Linux上的地位就仿佛Edit程序在DOS上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。

Vi不是一个排版程序，它不象Word或WPS那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。

Vi没有菜单，只有命令，且命令繁多。Vi有三种基本工作模式：命令行模式、文本输入模式和末行模式。

命令行模式

任何时候，不管用户处于何种模式，只要按一下键，即可使Vi进入命令行模式；我们在shell环境（提示符为$）下输入启动Vi命令，进入编辑器时，也是处于该模式下。

在该模式下，用户可以输入各种合法的Vi命令，用于管理自己的文档。此时从键盘上输入的任何字符都被当做编辑命令来解释，若输入的字符是合法的Vi命令，则Vi在接受用户命令之后完成相应的动作。但需注意的是，所输入的命令并不在屏幕上显示出来。若输入的字符不是Vi的合法命令，Vi会响铃报警。

文本输入模式

在命令模式下输入插入命令i、附加命令a 、打开命令o、修改命令c、取代命令r或替换命令s都可以进入文本输入模式。在该模式下，用户输入的任何字符都被Vi当做文件内容保存起来，并将其显示在屏幕上。在文本输入过程中，若想回到命令模式下，按键即可。

末行模式

末行模式也称ex转义模式。

Vi和Ex编辑器的功能是相同的，二者主要区别是用户界面。在Vi中，命令通常是单个键，例如i、a、o等；而在Ex中，命令是以按回车键结束的正文行。Vi有一个专门的“转义”命令，可访问很多面向行的Ex命令。在命令模式下，用户按“:”键即可进入末行模式下，此时Vi会在显示窗口的最后一行（通常也是屏幕的最后一行）显示一个“:”作为末行模式的提示符，等待用户输入命令。多数文件管理命令都是在此模式下执行的（如把编辑缓冲区的内容写到文件中等）。末行命令执行完后，Vi自动回到命令模式。例如：

:1，$s / A / a / g

则从文件第一行至文件尾将大写A全部替换成小写a。

若在末行模式下输入命令过程中改变了主意，可按键，或用退格键将输入的命令全部删除之后，再按一下退格键，即可使Vi回到命令模式下。

Vi编辑器的三种工作模式之间的转换如图11-1所示。

如果要从命令模式转换到编辑模式，可以键入命令a或者i；如果需要从文本模式返回，则按Esc键即可。在命令模式下输入“:”即可切换到末行模式，然后输入命令。

第二十五课 Vi的进入与退出　　　　　2000年/6月/13日

Vi 的进入

用户登录到系统中之后，系统给出提示符“$”。在提示符后键入Vi和想要编辑（或建立）的文件名，便可进入Vi。

［例1］ 键入命令：

$ vi example.c

屏幕显示如下：

~

~

~

~

~

~

~

~

~

~

~

~

~

“ example.c” ［New File］

如果只键入Vi，而不带文件名，也可以进入Vi。之后在光标处键入文件内容，退出Vi时，只需在退出命令后输入文件名即可。

进入Vi之后，首先进入的就是命令模式，也就是说等待命令输入而不是文本输入。这时输入的字母都将作为命令来解释。光标停在屏幕第一行首位上（用表示），其余各行行首均有一个“~”符号，表示该行为空行。最后一行也称状态行，显示出当前正在编辑的文件名以及其状态。如本例是［New File］，表示example.c是一个新建的文件。如果example.c文件已在系统中存在，那么输入上述命令后，则在屏幕上显示出该文件的内容，并且光标停在第一行的首位，在状态行显示出该文件的文件名、行数和字符数。

［例2］ 键入命令：

$ vi example.c

然后键入：

```
#include

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）;

}

add（ ）

{ static int x=0;

x++ ;

printf（“x = %d\n”， x）;

}


~

~

~

~
```

“ example.c” ［ New File］

注意，在这里我们为了节省篇幅起见，屏幕只显示15行。事实上，初始的显示行数与用户所用终端有关，一般的CRT终端可显示25行。在窗口系统中，显示行数与运行Vi的那个窗口有关。然而我们可以对显示行数进行设置。例如，在能显示25行的CRT终端上，让Vi只显示15行。设置的方法将在本节后面介绍。

当用Vi建立一个新文件时，在进入Vi的命令中也可以不给出文件名，当编辑完文件需要保存数据时，再由用户指定文件名。

进入Vi时，用户不仅可以指定一个待编辑的文件名，而且还有许多附加操作。

如果希望在进入Vi之后，光标处于文件中特定的某行上，可在Vi后加上任选项+n，其中n为指定的行数。

［例3］ 键入命令：

$ vi +5 example1.c

后，屏幕显示如下：

```
#include

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）;

}

add（ ）

{ static int x=0;

x++ ;

printf（“x = %d\n”, x）;

}

~

~
```

~

~

~

“ example.c” 12 lines, 125 characters

光标将位于文件example1.c中的第5行上。

如果希望在进入Vi之后光标处于文件最末行，则只需把命令中附加项“+”后面的数字n省略掉即可。

在进入Vi时，除了可以指定一个光标起始行号之外，还可以在命令中指定一个模式串，此时在进入Vi后，光标就处于文件中第一个与指定模式串相匹配的那行上。

［例4］ 键入命令：

$ vi +/int example1.c

屏幕显示如下：

```
#include

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）;

}

add（ ）

{ static int x=0;

x++ ;

printf（“x = %d\n”, x）;

}

~

~

~
```

~

~

“ example.c” 12 lines, 125 characters

光标将位于文件example1.c中的第3行上。

使用Vi可以同时编辑多个文件，只要在进入Vi的命令中写入所要操作的文件即可，还可以使用通配符。

［例5］ 键入命令：

$ vi *.cat

就可以编辑所有后缀为cat的文件了。当然，您还可以使用shell中内置的模式匹配来装载文件，这样会更富技巧性。一旦您在命令行中包含了所有的待编辑文件名，就可以在Vi中方便地移来移去。

退出Vi

当编辑完文件，准备退出Vi返回到shell时，可以使用以下几种方法之一。

在命令模式中，连按两次大写字母Z，若当前编辑的文件曾被修改过，则Vi保存该文件后退出，返回到shell；若当前编辑的文件没被修改过，则Vi直接退出, 返回到shell。

在末行模式下，输入命令

:w

Vi保存当前编辑文件，但并不退出，而是继续等待用户输入命令。在使用w命令时，可以再给编辑文件起一个新的文件名。

［例6］

:w newfile

此时Vi将把当前文件的内容保存到指定的newfile中，而原有文件保持不变。若newfile是一个已存在的文件，则Vi在显示窗口的状态行给出提示信息：

File exists （use ! to override）

此时，若用户真的希望用文件的当前内容替换newfile中原有内容，可使用命令

:w! newfile

否则可选择另外的文件名来保存当前文件。

在末行模式下，输入命令

:q

系统退出Vi返回到shell。若在用此命令退出Vi时，编辑文件没有被保存，则Vi在显示窗口的最末行显示如下信息：

No write since last change （use ! to overrides）

提示用户该文件被修改后没有保存，然后Vi并不退出，继续等待用户命令。若用户就是不想保存被修改后的文件而要强行退出Vi时，可使用命令

:q!

Vi放弃所作修改而直接退到shell下。

在末行模式下，输入命令

:wq

Vi将先保存文件，然后退出Vi返回到shell。

在末行模式下，输入命令

该命令的功能同命令模式下的ZZ命令功能相同。[/b]

碧轩
人之初，性本色!
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

最后进行编辑的是 bixuan on 2003-04-12, 9:46am, 总计第 1 次编辑

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:08am 发表主题:

Vi中的行号

Vi中的许多命令都要用到行号及行数等数值。若编辑的文件较大时，自己去数是非常不方便的。为此Vi提供了给文本加行号的功能。这些行号显示在屏幕的左边，而相应行的内容则显示在行号之后。

使用的命令为：在末行方式下输入命令：

:set number

需要说明的是，这里加的行号只是显示给用户看的，它们并不是文件内容的一部分。

在一个较大的文件中，用户可能需要了解光标当前行是哪一行，在文件中处于什么位置，可在命令模式下用组合键<Ctrl+g>，此时Vi会在显示窗口的最后一行显示出相应信息。该命令可以在任何时候使用。

［例7］

```
#include <stdio.h>

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）;

}

add（ ）

{ static int x=0;

x++ ;

printf（“x = %d\n”, x）;
```

}

~

~

~

~

~

“ example.c” ［Modified］ lines 4 of 10 --40%-- col 11

在末行方式下，我们可以输入命令nu（单词number的缩写）来获得光标当前行的行号与该行内容。

光标移动操作

全屏幕文本编辑器中，光标的移动操作无疑是最经常使用的操作了。用户只有熟练地使用移动光标的这些命令，才能迅速准确地到达所期望的位置处进行编辑。

Vi中的光标移动既可以在命令模式下，也可以在文本输入模式下，但操作的方法不尽相同。

在文本输入模式下，可直接使用键盘上的四个方向键移动光标。

在命令模式下，有很多移动光标的方法。不但可以使用四个方向键来移动光标，还可以用h、j、k、l这四个键代替四个方向键来移动光标，这样可以避免由于不同机器上的不同键盘定义所带来的矛盾，而且使用熟练后可以手不离开字母键盘位置就能完成所有操作，从而提高工作效率。另外还可以用<Spacebar>、<Backspace>、<Ctrl+n>和<ctrl+p>四个键或组合键移动光标。且以上这三种键在实现功能上是等价的。除此之外，还有一些移动光标的命令。下面对它们的工作方式介绍如下：

<Spacebar>、® （均称为右向键）

右向键的作用是将光标向右移动一个位置。若在向右键前先输入一个数字n，那么光标就向右移动n个位置。例如5l表示光标向右移动5个位置。需要注意的是，光标移动不能超过当前行的末尾。若给定的n超过光标当前位置至行尾的字符个数，如果用右向键，光标只能移到行尾；如果用<Spacebar>，光标移到下面一行或几行的适当位置。

h、<Backspace>、¬ （向左键）

执行一次向左键，光标向左移动一个位置。同向右键一样，也可以在向左键的前面输入一个数字n，那么光标就向左移动n个位置。需要注意的是，如果用左向键，光标左移不能超出该行的开头；如果用<Backspace>，光标移到上面一行或几行的适当位置。

j、<Ctrl+n>、¯ （向下键）

执行一次向下键光标向下移动一个位置（即一行），但光标所在的列不变。当这些命令前面加上数字n，

则光标下移n行。

Vi除了可以用向下键将光标下移外，还可以用<Enter>键和“+”键将光标下移一行或n行（不包括本行在内），但此时光标下移之后将位于该行的第一个字符处。例如：

3j 光标下移3行，且光标所在列的位置不变。

3+或3<Enter> 光标下移3行，且光标位于该行的行首。

k、<Ctrl+p>、 （向上键）

执行一次向上键光标向上移动一个位置（即一行），但光标所在的列不变。同样在这些命令前面加上数字n，则光标上移n行。

若希望光标上移之后，光标位于该行的行首，则可以使用命令“- ”。

L （移至行首）

L 命令是将光标移到当前行的开头，即将光标移至当前行的第一个非空白处（非制表符或非空格符）。

$（移至行尾）

该命令将光标移到当前行的行尾，停在最后一个字符上。若在$命令之前加上一个数字n，则光标下移n-1行并到达行尾。

［行号］ G（移至指定行）

该命令将光标移至指定行号所指定的行的行首。这种移动称为绝对定位移动。

［例8］

```
#include <stdio.h>

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）;

}

add（ ）

{ static int x=0;

x++ ;
```

```
printf (“x = %d\n”, x);

}

~

~

~

~

~

“ example.c” 12 lines, 125 characters
```

键入命令:6G后，屏幕显示如下：

```
#include <stdio.h>

main ( )

{ int k ;

for ( k=0 ; k<3 ; k++) add ( ) ;

}

add ( )

{ static int x=0;

x++ ;

printf (“x = %d\n”, x);

}

~

~

~

~

~
```

“ example.c” 12 lines, 125 characters

光标移到了第6行的行首。

若省略行号，则光标移至该文件的最后一行的行首，即无论该文件有多少屏，都跳至最后一行。

第二十七课 vi命令　　2000年/6月/26日

在屏幕上移动

Vi提供了三个关于光标在全屏幕上移动并且文件本身不发生滚动的命令。它们分别是H、M和L命令。

1. H命令

该命令将光标移至屏幕首行的行首（即左上角），也就是当前屏幕的第一行，而不是整个文件的第一行。利用此命令可以快速将光标移至屏幕顶部。若在H命令之前加上数字n，则将光标移至第n行的行首。

［例9］ 屏幕显示如下内容：

```
#include <stdio.h>

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）; }

add（ ）

{ static int x=0;

x++ ;

printf（“x=%d\n”, x）;

}

~

~

~
```

~

~

“ example.c” ［ New File］

在命令模式下输入4H命令后，光标移到以for开头的这一行的字母f上。

值得一提的是，使用命令dH将会删除从光标当前所在行至所显示屏幕首行的全部内容。

2. M命令

该命令将光标移至屏幕显示文件的中间行的行首。即如果当前屏幕已经充满,则移动到整个屏幕的中间行；如果并未充满，则移动到文本的那些行的中间行。利用此命令可以快速地将光标从屏幕的任意位置移至屏幕显示文件的中间行的行首。例如，在上面屏幕显示的情况下（不论光标在屏幕的何处），在命令模式下，输入命令M之后，光标都将移到add这一行的字母a上。

同样值得一提的是，使用命令dM将会删除从光标当前所在行至屏幕显示文件的中间行的全部内容。

3. L命令

当文件显示内容超过一屏时，该命令将光标移至屏幕上的最底行的行首；当文件显示内容不足一屏时，该命令将光标移至文件的最后一行的行首。可见，利用此命令可以快速准确地将光标移至屏幕底部或文件的最后一行。若在L命令之前加上数字n，则将光标移至从屏幕底部算起第n行的行首。例如，在上面屏幕显示的情况下（不论光标在屏幕的何处），在命令模式下，输入命令3L之后，光标都将移到x++这一行的字母x上。

同样值得一提的是，使用命令dL将会删除从光标当前行至屏幕底行的全部内容。

按字移动光标

首先介绍一下Vi中“字”的概念。在Vi中“字”有两种含义。一种是广义的字，它可以是两个空格之间的任何内容。

［例10］ 在文件example.c 中的第4行：

{int k；

它只有2个字，一个是{int ，另一个是k;

Vi中另一种字是狭义上的字，在此种意义之下，英文单词、标点符号和非字母字符（如!、@、#、$、%、^、&、*、（、）、-、+、{、}、［、］、~、|、\、<、>、/等）均被当成是一个字。因此，上面那一行中就包括{、int、k、；这4个字。

Vi中使用大写命令一般就是指将字作为广义来对待，使用小写命令就是作为狭义对待。

搞清楚Vi中字的含义后，我们就可以介绍按字移动光标的命令了。

Vi 一共提供了三组关于按字移动光标的命令，分别是：

1. w和W命令

将光标右移至下一个字的字首；

［例11］ 屏幕上显示如下：

printf（“Hello Mr.Huang!\n”）;

现在使用w命令，把光标移到下一个字（狭义）的字首“（”上：

printf（“Hello Mr.Huang!\n”）;

下面使用W命令，将把光标移到下一个字（广义）的字首“M”上：

printf（“Hello Mr.Huang!\n”）;

2. e和E命令

如果光标起始位置处于字内（即非字尾处），则该命令将把光标移到本字字尾；如果光标起始位置处于字尾，则该命令将把光标移动到下一个字的字尾。

3. b和B

如果光标处于所在字内（即非字首），则该命令将把光标移至本字字首；如果光标处于所在字字首，则该命令将把光标移到上一个字的字首。

［例12］ 屏幕显示如下：

printf（“Hello Mr.Huang!\n”）;

现在使用b命令，由于光标处于该字中间，所以光标移动到本字字首“H”处：

printf（“Hello Mr.Huang!\n”）;

如果使用B命令，则如下所示：

printf（“Hello Mr.Huang!\n”）;

按句移动光标

在Vi中，一个句子被定义为是以逗号（,）、句号（.）、问号（?）和感叹号（!）结尾，且其后面跟着至少两个以上（含两个）空格或一个换行符的字符序列。

Vi提供了关于按句移动光标的两个命令，分别为：

1. （ 命令

将光标移至上一个句子的开头。

2. ） 命令

该命令将光标移至下一个句子的开头。

按段移动光标

在Vi中，一个段被定义为是以一个空白行开始和结束的片段。Vi提供了关于按段移动光标的两个命令，分别为：

1. { 命令

该命令将光标向前移至上一个段的开头；

2. } 命令

该命令将光标向后移至下一个段的开头。

屏幕滚动

屏幕命令是以屏幕为单位移动光标的，常用于文件的滚屏和分页。需要注意的是，屏幕命令不是光标移动命令，不能作为文本限定符用于删除命令中。

在命令模式下和文本输入模式下均可以使用屏幕滚动命令。

1. 滚屏命令

关于滚屏命令有两个：

- < Ctrl+u > 将屏幕向前（文件头方向）翻滚半屏；

- < Ctrl+d > 将屏幕向后（文件尾方向）翻滚半屏。

可以在这两个命令之前加上一个数字n，则屏幕向前或向后翻滚n行。并且这个值被系统记住，以后再用< Ctrl+u >和< Ctrl+d >命令滚屏时，还滚相应的行数。

2. 分页命令

关于分页命令也有两个：

- < Ctrl+f > 将屏幕向文件尾方向翻滚一整屏（即一页）；

- < Ctrl+b > 将屏幕向文件首方向翻滚一整屏（即一页）。

同样也可以在这两个命令之前加上一个数字n，则屏幕向前或向后移动n页。

3. 状态命令< Ctrl+G >

命令显示在vi状态行上的vi状态信息，包括正在编辑的文件名、是否修改过、当前行号、文件的行数以及光标之前的行占整个文件的百分比。

4. 屏幕调零命令

Vi 提供了三个有关屏幕调零的命令。它们的格式分别为：

- ［行号］ z ［行数］ <回车>
- ［行号］ z ［行数］ .
- ［行号］ z ［行数］ _

若省略了行号和行数，这三个命令分别为将光标所在的当前行作为屏幕的首行、中间行和最末行重新显示；若给出行号，那么该行号所对应的行就作为当前行显示在屏幕的首行、中间行和最末行；若给出行数，则它规定了在屏幕上显示的行数。

［例13］

8z16<回车> ：将文件中的第8行作为屏幕显示的首行，并一共显示16行。

15z . ：将文件中的第15行作为屏幕显示的中间行，显示行数为整屏。

15z 5_ ：将文件中的第15行作为屏幕显示的最末行，显示行数为5行。

文本插入操作

在命令模式下用户输入的任何字符都被Vi当作命令加以解释执行，如果用户要将输入的字符当作是文本内容时，则首先应将Vi的工作模式从命令模式切换到文本输入模式。切换的方式是使用下面的命令。

插入（Insert）命令

Vi提供了两个插入命令：i和 I。

1. i命令

插入文本从光标所在位置前开始,并且插入过程中可以使用<Backspace>键删除错误的输入。此时Vi处于插入状态，屏幕最下行显示“--INSERT--”（插入）字样。

［例14］ 有一正在编辑的文件，如下所示：

Welcome to vi world!Come on!

~

~

光标位于第一个“!”上，需在其前面插入：

This is an example!

使用i命令，并输入相应文本后，屏幕显示如下：

Welcome to vi world This is an example!!Come on!

~

~

由此例可以看到，光标本来是在第一个“!”处，但是由于是从光标所在位置前开始插入，所以这个“!”就被挤到了新插入的文本之后。

2. I命令

该命令是将光标移到当前行的行首，然后在其前插入文本。

____

碧轩

人之初，性本色！

http://www.ourlinux.net/

http://yum.ourlinux.net/

http://apt.ourlinux.net/

http://download.ourlinux.net/

游戏交易：http://www.0579tt.com/

返回页首

bixuan

公社特别嘉宾

注册: Nov 23, 2002

文章: 1994

位置: 浙江金华

发表于: 2003-04-12, 9:09am 发表主题:

----

vi命令

在屏幕上移动

Vi提供了三个关于光标在全屏幕上移动并且文件本身不发生滚动的命令。它们分别是H、M和L命令。

1. H命令

该命令将光标移至屏幕首行的行首（即左上角），也就是当前屏幕的第一行，而不是整个文件的第一行。利用此命令可以快速将光标移至屏幕顶部。若在H命令之前加上数字n，则将光标移至第n行的行首。

［例9］ 屏幕显示如下内容：

```
#include <stdio.h>

main （ ）

{ int k ;

for （ k=0 ; k<3 ; k++） add（ ）; }

add（ ）

{ static int x=0;

x++ ;

printf（“x=%d\n”, x）;

}

~

~

~

~

~

“ example.c” ［ New File］
```

在命令模式下输入4H命令后，光标移到以for开头的这一行的字母f上。

值得一提的是，使用命令dH将会删除从光标当前所在行至所显示屏幕首行的全部内容。
2. M命令

该命令将光标移至屏幕显示文件的中间行的行首。即如果当前屏幕已经充满,则移动到整个屏幕的中间行；如果并未充满，则移动到文本的那些行的中间行。利用此命令可以快速地将光标从屏幕的任意位置移至屏幕显示文件的中间行的行首。例如，在上面屏幕显示的情况下（不论光标在屏幕的何处），在命令模式下，输入命令M之后，光标都将移到add这一行的字母a上。

同样值得一提的是，使用命令dM将会删除从光标当前所在行至屏幕显示文件的中间行的全部内容。
3. L命令

当文件显示内容超过一屏时，该命令将光标移至屏幕上的最底行的行首；当文件显示内容不足一屏时，该命令将光标移至文件的最后一行的行首。可见，利用此命令可以快速准确地将光标移至屏幕底部或文件的最后一行。若在L命令之前加上数字n，则将光标移至从屏幕底部算起第n行的行首。例如，在上面屏幕显示的情况下（不论光标在屏幕的何处），在命令模式下，输入命令3L之后，光标都将移到x++这一行的字母x上。

同样值得一提的是，使用命令dL将会删除从光标当前行至屏幕底行的全部内容。

按字移动光标

首先介绍一下Vi中“字”的概念。在Vi中“字”有两种含义。一种是广义的字，它可以是两个空格之间的任何内容。

［例10］ 在文件example.c 中的第4行：

{int k;

它只有2个字，一个是{int ，另一个是k;

Vi中另一种字是狭义上的字，在此种意义之下，英文单词、标点符号和非字母字符（如!、@、#、$、%、^、&、*、（、）、-、+、{、}、［、］、~、|、\、<、>、/等）均被当成是一个字。因此，上面那一行中就包括{、int、k、；这4个字。

Vi中使用大写命令一般就是指将字作为广义来对待，使用小写命令就是作为狭义对待。

搞清楚Vi中字的含义后，我们就可以介绍按字移动光标的命令了。

Vi 一共提供了三组关于按字移动光标的命令，分别是：

1. w和W命令

将光标右移至下一个字的字首；

［例11］ 屏幕上显示如下：

printf（“Hello Mr.Huang!\n”）;

现在使用w命令，把光标移到下一个字（狭义）的字首“（”上：

printf（“Hello Mr.Huang!\n”）;

下面使用W命令，将把光标移到下一个字（广义）的字首“M”上：

printf（“Hello Mr.Huang!\n”）;
2. e和E命令

如果光标起始位置处于字内（即非字尾处），则该命令将把光标移到本字字尾；如果光标起始位置处于字尾，则该命令将把光标移动到下一个字的字尾。
3. b和B

如果光标处于所在字内（即非字首），则该命令将把光标移至本字字首；如果光标处于所在字字首，则该命令将把光标移到上一个字的字首。

［例12］ 屏幕显示如下：

printf（“Hello Mr.Huang!\n”）;

现在使用b命令，由于光标处于该字中间，所以光标移动到本字字首“H”处：

printf（“Hello Mr.Huang!\n”）;

如果使用B命令，则如下所示：

printf（“Hello Mr.Huang!\n”）;

按句移动光标

在Vi中，一个句子被定义为是以逗号（,）、句号（.）、问号（?）和感叹号（!）结尾，且其后面跟着至少两个以上（含两个）空格或一个换行符的字符序列。

Vi提供了关于按句移动光标的两个命令，分别为：

1. （ 命令

将光标移至上一个句子的开头。

2. ） 命令

该命令将光标移至下一个句子的开头。

按段移动光标

在Vi中，一个段被定义为是以一个空白行开始和结束的片段。Vi提供了关于按段移动光标的两个命令，分别为：

1. { 命令

该命令将光标向前移至上一个段的开头；

2. } 命令

该命令将光标向后移至下一个段的开头。

屏幕滚动

屏幕命令是以屏幕为单位移动光标的，常用于文件的滚屏和分页。需要注意的是，屏幕命令不是光标移动命令，不能作为文本限定符用于删除命令中。

在命令模式下和文本输入模式下均可以使用屏幕滚动命令。

1. 滚屏命令

关于滚屏命令有两个：

- < Ctrl+u > 将屏幕向前（文件头方向）翻滚半屏；

- < Ctrl+d > 将屏幕向后（文件尾方向）翻滚半屏。

可以在这两个命令之前加上一个数字n，则屏幕向前或向后翻滚n行。并且这个值被系统记住，以后再用< Ctrl+u >和< Ctrl+d >命令滚屏时，还滚相应的行数。

2. 分页命令

关于分页命令也有两个：

- < Ctrl+f > 将屏幕向文件尾方向翻滚一整屏（即一页）；

- < Ctrl+b > 将屏幕向文件首方向翻滚一整屏（即一页）。

同样也可以在这两个命令之前加上一个数字n，则屏幕向前或向后移动n页。

3. 状态命令< Ctrl+G >

命令显示在vi状态行上的vi状态信息，包括正在编辑的文件名、是否修改过、当前行号、文件的行数以及光标之前的行占整个文件的百分比。

4. 屏幕调零命令

Vi 提供了三个有关屏幕调零的命令。它们的格式分别为：

- ［行号］ z ［行数］ <回车>
- ［行号］ z ［行数］ .
- ［行号］ z ［行数］ _

若省略了行号和行数，这三个命令分别为将光标所在的当前行作为屏幕的首行、中间行和最末行重新显示；若给出行号，那么该行号所对应的行就作为当前行显示在屏幕的首行、中间行和最末行；若给出行数，则它规定了在屏幕上显示的行数。

［例13］

8z16<回车> ：将文件中的第8行作为屏幕显示的首行，并一共显示16行。

15z . ：将文件中的第15行作为屏幕显示的中间行，显示行数为整屏。

15z 5_ ：将文件中的第15行作为屏幕显示的最末行，显示行数为5行。

文本插入操作

在命令模式下用户输入的任何字符都被Vi当作命令加以解释执行，如果用户要将输入的字符当作是文本内容时，则首先应将Vi的工作模式从命令模式切换到文本输入模式。切换的方式是使用下面的命令。

插入（Insert）命令

Vi提供了两个插入命令：i和 I。

1. i命令

插入文本从光标所在位置前开始,并且插入过程中可以使用<Backspace>键删除错误的输入。此时Vi处于插入状态，屏幕最下行显示“--INSERT--”（插入）字样。

［例14］ 有一正在编辑的文件，如下所示：

Welcome to vi world!Come on!

~

~

光标位于第一个“!”上，需在其前面插入：

This is an example!

使用i命令，并输入相应文本后，屏幕显示如下：

Welcome to vi world This is an example!!Come on!

~

~

由此例可以看到，光标本来是在第一个“!”处，但是由于是从光标所在位置前开始插入，所以这个“!”就被挤到了新插入的文本之后。
2. I命令

该命令是将光标移到当前行的行首，然后在其前插入文本。

}

第二十九课 vi命令　　　2000年/7/10日

附加（append）命令

Vi提供了两个附加插入命令：a和A。

1. a命令

该命令用于在光标当前所在位置之后追加新文本。新输入的文本放在光标之后，在光标后的原文本将相应地向后移动。光标可在一行的任何位置。

［例15］ 以例14原始情况为例，使用a命令，并输入相应文本，屏幕显示如下：

Welcome to vi world!This is an example!Come on!

~

~

本例中光标后的文本“Come on!”被新输入的文本挤到了后面。
2. A命令

该命令与a命令不同的是，A命令将把光标挪到所在行的行尾，从那里开始插入新文本。当输入A命令后，光标自动移到该行的行尾。

a和A命令是把文本插入到行尾的唯一方法。

打开（open）命令

不论是Insert命令也好，还是append命令也好，所插入的内容都是从当前行中的某个位置开始的。若我们希望在某行之前或某行之后插入一些新行，则应使用open命令。

Vi提供了两个打开命令：o和O。

1. o命令

该命令将在光标所在行的下面新开一行，并将光标置于该行的行首，等待输入文本。要注意，当使用<Backspace>删除字符时只能删除从插入模式开始的位置以后的字符，对于以前的字符不起作用。而且还可以在文本输入方式下输入一些控制字符，例如，Ctrl+l即是插入分页符，显示为^L。

［例16］ 以例14的原始情况为例，只是这次要在当前行下面一行输入文本。使用o命令，并输入相应文本，屏幕显示如下：

Welcome to vi world!Come on!

This is an example!

~

~

新输入的文本出现在原来文本所在行的下一行。
2. O命令

和o命令相反，O命令是在光标所在行的上面插入一行，并将光标置于该行的行首，等待输入文本。

［例17］ 与例14的原始情况相同，要在当前行的上面一行输入文本。使用O命令并输入文本后，屏幕显示如下：

This is an example!

Welcome to vi world!Come on!

~

~

文本插入到了当前行之上。

11.3.4　文本修改

在命令模式下可以使用Vi提供的各种有关命令对文本进行修改，包括对文本内容的删除、复制、取代和替换等。

文本删除

在编辑文本时，经常需要删除一些不需要的文本，我们可以用<Backspace>键将输错或不需要的文本删除，但此时有一个限制就是当删到行头之后，再想删上面那行的内容是不可能的。

在命令模式下，Vi提供了许多删除命令。这些命令大多是以d开头的。常用的有：

1. 删除单个字符

x：　删除光标处的字符。若在x之前加上一个数字n，则删除从光标所在位置开始向右的n个字符。

X：删除光标前面的那个字符。若在X之前加上一个数字n，则删除从光标前面那个字符开始向左的n个字符。

显然这两个命令是删除少量字符的快捷方法。

2. 删除多个字符

dd：删除光标所在的整行。在dd前可加上一个数字n，表示删除当前行及其后n-1行的内容。

D或d$：两命令功能一样，都是删除从光标所在处开始到行尾的内容。

d0：删除从光标前一个字符开始到行首的内容。

dw：删除一个单词。若光标处在某个词的中间，则从光标所在位置开始删至词尾。同dd命令一样，可在dw之前加一个数字n，表示删除n个指定的单词。

如果用户不小心进行了误删除操作，也不要紧，Vi提供了恢复误操作的命令，并且可以将恢复的内容移动，放在文本的任何地方。恢复命令用² np，其中n为寄存器号。这是因为Vi内部有9个用于维护删除操作的寄存器，分别用数字1，2，¼ ，9表示，它们分别保存以往用dd命令删除的内容。这些寄存器组成一个队列，例如最近一次使用dd命令删除的内容被放到寄存器1中；当下次再使用dd命令删除文本内容时，Vi将把寄存器1的内容转存到寄存器2中，而寄存器1中又将是最近一次dd命令删除的内容。

以此类推，Vi可以保存有最近九次用dd命令删除的内容，而前面的用dd命令删除的内容则被抛弃。

［例18］ 假设当前编辑文件为xu.c

```
/* this is a example */

#include <stdio.h>

void main（ ）

{

int i ， j ;

printf（ “ please input a number : / n ” ）;

scanf （ “ % d ” ，&i ） ;

j = i + 100 ;

printf （ “ /n j = % d /n ” ， j ） ;

return ;

}
```

我们对其进行如下操作：

1. 将光标移至文件第一行，按dd命令，此时文件第一行的内容被删除，且被删除的内容保存在寄存器1中；
2. 按5j使光标下移至第一个printf语句行；
3. 按dd命令将该行删除，此时寄存器1中将保存刚刚被删除的内容：

```
printf （“ please input a number :\ n ” ）;
```

而寄存器1原有的内容：

```
/* this is a example */
```

则被保存到寄存器2中；

在最末行模式下，也可以对文件内容进行删除，但它只能删除整行，一次可将某个指定范围内（起始行号，终止行号）的所有行全部删除。需要注意的是，用此种方法进行删除时，Vi并不把所删内容放入寄存器中，因而当发生误删除操作时，不能用² np命令恢复，只能用u命令进行有限的恢复。

最后提一下，如何在文本输入方式时将所输入文本删除。用户使用<Ctrl+u>组合键即可，此时光标将返回插入开始的位置，并且Vi仍处于文本输入方式。

取消上一命令（Undo）

取消上一命令（Undo），也称复原命令，是非常有用的命令，它可以取消前一次的误操作或不合适的操作对文件造成的影响，使之回复到这种误操作或不合适操作被执行之前的状态。

取消上一命令有两种形式，在命令模式下键入字符u和U。它们的功能都是取消刚才输入的命令，恢复到原来的情况。小写u和大写U在具体细节上有所不同，二者的区别在于，大写U命令的功能是恢复到误操作命令前的情况，即如果插入命令后使用U命令，就删除刚刚插入的内容；如果删除命令后使用U命令，就相当于在光标处又插入刚刚删除的内容。这里把所有修改文本的命令都视为插入命令。也就是说，U命令只能取消前一步操作，如果用U命令撤消了前一步操作，当再按U键时，并不是撤消再前一步的操作，而是撤消了刚才U命令执行的操作，也就是又恢复到第一次使用U命令之前的状态，结果是什么都没做。而小写u命令的功能是把当前行恢复成被编辑前的状态，而不管此行被编辑了多少次。

［例19］ 原来屏幕显示内容为：

```
#include <stdio.h>

main （ ）

{

}
```

在命令模式下输入命令o，插入一新行，输入需要插入的内容后再按<Esc>回到命令模式，屏幕显示内容为：

```
#include <stdio.h>

main （ ）

{

printf （ “ How do you do ! ” ） ;

}
```

若想取消这一插入操作，请按命令U后，屏幕恢复到原来显示的情况。

注意：对于取消命令仍可以再使用取消命令。这时会产生一种“负负得正”的效果，文件状态将恢复到第一次执行取消命令之前的状态，如同没做任何操作一般。例如在上例中，再使用一次命令U，屏幕将显示的内容仍为插入后的内容。

重复命令（Redo）

重复命令也是一个非常常用的命令。在文本编辑中经常会碰到需要机械地重复一些操作，这时就需要用到重复命令。它可以让用户方便地再执行一次前面刚完成的某个复杂的命令。

重复命令只能在命令模式下工作，在该模式下按“.”键既可。执行一个重复命令时，其结果是依赖于光标当前位置的。

［例20］ 屏幕显示内容为：

```
#include <stdio.h>

main （ ）

{

}
```

输入命令o，并输入一行内容后，再< Esc >返回到命令模式下，屏幕显示内容为：

```
#include <stdio.h>

main （ ）

{

printf （ “ How do you do ! ” ） ;

}
```

此时输入命令“.”，屏幕显示内容为：

```
#include <stdio.h>

main （ ）

{

printf （ “ How do you do ! ” ） ;

printf （ “ How do you do ! ” ） ;
```

碧轩
人之初，性本色！
http://www.ourlinux.net/

http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:14am 发表主题:

----

shell命令[转帖]
shell是用户和Linux操作系统之间的接口。Linux中有多种shell，其中缺省使用的是Bash。本章讲述了shell的工作原理，shell的种类，shell的一般操作及Bash的特性。

什么是shell

Linux系统的shell作为操作系统的外壳，为用户提供使用操作系统的接口。它是命令语言、命令解释程序及程序设计语言的统称。

shell是用户和Linux内核之间的接口程序，如果把Linux内核想象成一个球体的中心，shell就是围绕内核的外层。当从shell或其他程序向Linux传递命令时，内核会做出相应的反应。

shell是一个命令语言解释器，它拥有自己内建的shell命令集，shell也能被系统中其他应用程序所调用。用户在提示符下输入的命令都由shell先解释然后传给Linux核心。

有一些命令，比如改变工作目录命令cd，是包含在shell内部的。还有一些命令，例如拷贝命令cp和移动命令rm，是存在于文件系统中某个目录下的单独的程序。对用户而言，不必关心一个命令是建立在shell内部还是一个单独的程序。

shell首先检查命令是否是内部命令，若不是再检查是否是一个应用程序（这里的应用程序可以是Linux本身的实用程序，如ls和rm，也可以是购买的商业程序，如xv，或者是自由软件，如emacs）。然后shell在搜索路径里寻找这些应用程序（搜索路径就是一个能找到可执行程序的目录列表）。如果键入的命令不是一个内部命令并且在路径里没有找到这个可执行文件，将会显示一条错误信息。如果能够成功找到命令，该内部命令或应用程序将被分解为系统调用并传给Linux内核。

shell的另一个重要特性是它自身就是一个解释型的程序设计语言，shell程序设计语言支持绝大多数在高级语言中能见到的程序元素，如函数、变量、数组和程序控制结构。shell编程语言简单易学，任何在

提示符中能键入的命令都能放到一个可执行的shell程序中。

当普通用户成功登录，系统将执行一个称为shell的程序。正是shell进程提供了命令行提示符。作为默认值（TurboLinux系统默认的shell是BASH），对普通用户用“$”作提示符，对超级用户（root）用“#”作提示符。

一旦出现了shell提示符，就可以键入命令名称及命令所需要的参数。shell将执行这些命令。如果一条命令花费了很长的时间来运行，或者在屏幕上产生了大量的输出，可以从键盘上按ctrl+c发出中断信号来中断它（在正常结束之前，中止它的执行）。

当用户准备结束登录对话进程时，可以键入logout命令、exit命令或文件结束符（EOF）（按ctrl+d实现），结束登录。

我们来实习一下shell是如何工作的。

$ make work

make:***No rule to make target ‘work’. Stop.

$

注释：make是系统中一个命令的名字，后面跟着命令参数。在接收到这个命令后，shell便执行它。本例中，由于输入的命令参数不正确，系统返回信息后停止该命令的执行。

在例子中，shell会寻找名为make的程序，并以work为参数执行它。make是一个经常被用来编译大程序的程序，它以参数作为目标来进行编译。在“make work”中，make编译的目标是work。因为make找不到以work为名字的目标，它便给出错误信息表示运行失败，用户又回到系统提示符下。

另外，用户键入有关命令行后，如果shell找不到以其中的命令名为名字的程序，就会给出错误信息。例如，如果用户键入：

$ myprog

bash:myprog:command not found

$

可以看到，用户得到了一个没有找到该命令的错误信息。用户敲错命令后，系统一般会给出这样的错误信息。

shell的种类

Linux中的shell有多种类型，其中最常用的几种是Bourne shell（sh）、C shell（csh）和Korn shell（ksh）。三种shell各有优缺点。Bourne shell是UNIX最初使用的shell，并且在每种UNIX上都可以使用。Bourne shell在shell编程方面相当优秀，但在处理与用户的交互方面做得不如其他几种shell。Linux操作系统缺省的shell是Bourne Again shell，它是Bourne shell的扩展，简称Bash，与Bourne shell完全向后兼容，并且在Bourne shell的基础上增加、增强了很多特性。Bash放在/bin/bash中，它

有许多特色，可以提供如命令补全、命令编辑和命令历史表等功能，它还包含了很多C shell和Korn shell中的优点，有灵活和强大的编程接口，同时又有很友好的用户界面。

C shell是一种比Bourne shell更适于编程的shell，它的语法与C语言很相似。 Linux为喜欢使用C shell的人提供了Tcsh。Tcsh是C shell的一个扩展版本。Tcsh包括命令行编辑、可编程单词补全、拼写校正、历史命令替换、作业控制和类似C语言的语法，它不仅和Bash shell是提示符兼容，而且还提供比Bash shell更多的提示符参数。

Korn shell集合了C shell和Bourne shell的优点并且和Bourne shell完全兼容。Linux系统提供了pdksh（ksh的扩展），它支持任务控制，可以在命令行上挂起、后台执行、唤醒或终止程序。

Linux并没有冷落其他shell用户，还包括了一些流行的shell如ash、zsh等。每个shell都有它的用途，有些shell是有专利的，有些能从Internet网上或其他来源获得。要决定使用哪个shell，只需读一下各种shell的联机帮助，并试用一下。

用户在登录到Linux时由/etc/passwd文件来决定要使用哪个shell。例如：

fgrep lisa /etc/passwd

lisa:x:500:500:TurboLinux User:/home/lisa:/bin/bash

shell被列每行的末尾（/bin/bash）。

由于Bash是Linux上缺省的shell，本章主要介绍Bash及其相关知识。

shell命令

命令行c

用户登录到Linux系统时，可以看到一个shell提示符，标识了命令行的开始。用户可以在提示符后面输入任何命令及参数。例如：

$ date

二 11 23 01:34:58 CST 1999

$

用户登录时，实际进入了shell，它遵循一定的语法将输入的命令加以解释并传给系统。命令行中输入的第一个字必须是一个命令的名字，第二个字是命令的选项或参数，命令行中的每个字必须由空格或TAB隔开，格式如下：

$ Command Option Arguments

1. 选项和参数

选项是包括一个或多个字母的代码，它前面有一个减号（减号是必要的，Linux用它来区别选项和参

数），选项可用于改变命令执行的动作的类型。例如：

$ ls

motd passwd

$

这是没有选项的ls命令，可列出当前目录中所有文件，只列出各个文件的名字，而不显示其他更多的信息。

$ ls -l

total 2

-rw-r--r-- 2 wzh book 22 Apr 20 20:37 motd

-rw-r--r-- 2 wzh book 796 Apr 20 20:37 passwd

$

加入-l选项，将会为每个文件列出一行信息，诸如数据大小和数据最后被修改的时间。

大多数命令都被设计为可以接纳参数。参数是在命令行中的选项之后键入的一个或多个单词，例如：

$ ls -l text

-rw-r--r-- 2 wzh book 22 Apr 20 20:37 motd

-rw-r--r-- 2 wzh book 796 Apr 20 20:37 passwd

$

将显示text目录下的所有文件及其信息。

有些命令，如ls可以带参数，而有一些命令可能需要一些最小数目的参数。例如，cp命令至少需要两个参数，如果参数的数目与命令要求不符，shell将会给出出错信息。例如：

$ cp -i mydata newdata

注意：命令行中选项先于参数输入。
2. 命令行特征

命令行实际上是可以编辑的一个文本缓冲区，在按回车之前，可以对输入的文本进行编辑。比如利用BACKSPACE键可以删除刚键入的字符，可以进行整行删除，还可以插入字符，使得用户在输入命令，尤其是复杂命令时，若出现键入错误，无须重新输入整个命令，只要利用编辑操作，即可改正错误。

利用上箭头可以重新显示刚执行的命令，利用这一功能可以重复执行以前执行过的命令，而无须重新键入该命令。

bash保存着以前键入过的命令的列表，这一列表被称为命令历史表。按动上箭头，便可以在命令行上逐次显示各条命令。同样，按动下箭头可以在命令列表中向下移动，这样可以将以前的各条命令显示在命令行上，用户可以修改并执行这些命令。这一特征将在10.4节中进行详细的论述。

在一个命令行中还可以置入多个命令，用分号将各个命令隔开。例如：

$ ls -F;cp -i mydata newdata

也可以在几个命令行中输入一个命令，用反斜杠将一个命令行持续到下一行。

$ cp －i

mydata

newdata

上面的cp命令是在三行中输入的，开始的两行以反斜杠结束，把三行作为一个命令行。

shell中的特殊字符

shell中除使用普通字符外，还可以使用一些具有特殊含义和功能的特殊字符。在使用它们时应注意其特殊的含义和作用范围。下面分别对这些特殊字符加以介绍。

1. 通配符

通配符用于模式匹配，如文件名匹配、路经名搜索、字符串查找等。常用的通配符有*、?和括在方括号［ ］中的字符序列。用户可以在作为命令参数的文件名中包含这些通配符，构成一个所谓的“模式串”，在执行过程中进行模式匹配。

* 代表任何字符串（长度可以不等），例如：“f*”匹配以f打头的任意字符串。但应注意，文件名前的圆点（.）和路经名中的斜线（/）必须显式匹配。例如“*”不能匹配.file，而“.*”才可以匹配.file。

? 代表任何单个字符。

［］ 代表指定的一个字符范围，只要文件名中［ ］位置处的字符在［ ］中指定的范围之内，那么这个文件名就与这个模式串匹配。方括号中的字符范围可以由直接给出的字符组成，也可以由表示限定范围的起始字符、终止字符及中间的连字符（-）组成。例如，f ［a- d］ 与f ［abcd］的作用相同。Shell将把与命令行中指定的模式串相匹配的所有文件名都作为命令的参数，形成最终的命令，然后再执行这个命令。

下面我们给出表10-1说明这些通配符的具体含义。

表10-1　通配符含义举例

模式串	意 义
*	当前目录下所有文件的名称。
Text	当前目录下所有文件名中包含有Text的文件的名称。
［ab-dm］＊	当前目录下所有以a、b、c、d、m开头的文件的名称。
［ab-dm］？	当前目录下所有以a、b、c、d、m开头且后面只跟有一个字符的文件的名称。
/usr/bin/??	目录/usr/bin下所有名称为两个字符的文件的名称。

特别需要注意的是，连字符“-”仅在方括号内有效，表示字符范围，如在方括号外面就成为普通字符了。而*和?只在方括号外面是通配符，若出现在方括号之内，它们也失去通配符的能力，成为普通字符了。例如，模式“- a［*?］abc”中只有一对方括号是通配符，*和?均为普通字符，因此，它匹配的字符串只能是- a*abc和- a?abc。

最后说明一下使用通配符时需要注意的一些问题。由于*、?和［ ］对于shell来说具有比较特殊的意义，因此在正常的文件名中不应出现这些字符。特别是在目录名中不要出现它们，否则Shell匹配起来可能会无穷的递归下去。另外要注意的一点是：如果目录中没有与指定的模式串相匹配的文件名，那么Shell将使用此模式串本身作为参数传给有关命令。这可能就是命令中出现特殊字符的原因所在。

2. 引号

在shell中引号分为三种：单引号，双引号和反引号。

* 单引号 ‘

由单引号括起来的字符都作为普通字符出现。特殊字符用单引号括起来以后，也会失去原有意义，而只作为普通字符解释。例如：

$ string=’$PATH’

$ echo $string

$PATH

$

可见$保持了其本身的含义，作为普通字符出现。

* 双引号 “

由双引号括起来的字符，除$、、’、和”这几个字符仍是特殊字符并保留其特殊功能外，其余字符仍作为普通字符对待。对于$来说，就是用其后指定的变量的值来代替这个变量和$；对于而言，是转义字符，它告诉shell不要对其后面的那个字符进行特殊处理，只当作普通字符即可。可以想见，在双引号中需要在前面加上的只有四个字符$，，’和”本身。而对”号，若其前面没有加，则Shell会将它同前一个”号匹配。

例如，我们假定PATH的值为.:/usr/bin:/bin，输入如下命令：

$ TestString＝”$PATH\”$PATH”

$ echo $TestString

.:/usr/bin:/ bin”$PATH

$

读者可以自己试一下在第二个双引号之前不加会产生什么结果。

* 反引号 `

反引号（`）这个字符所对应的键一般位于键盘的左上角，不要将其同单引号（’）混淆。反引号括起来的字符串被shell解释为命令行，在执行时，shell首先执行该命令行，并以它的标准输出结果取代整个反引号（包括两个反引号）部分。例如：

$ pwd

/home/xyz

$ string=”current directory is `pwd`”

$ echo $string

current directour is /home/xyz

$

shell执行echo命令时，首先执行`pwd`中的命令pwd，并将输出结果/home/xyz取代`pwd`这部分，最后输出替换后的整个结果。

利用反引号的这种功能可以进行命令置换，即把反引号括起来的执行结果赋值给指定变量。例如：

$ today=`date`

$ echo Today is $today

Today is Mon Apr 15 16:20:13 CST 1999

$

反引号还可以嵌套使用。但需注意，嵌套使用时内层的反引号必须用反斜线（）将其转义。例如：

$ abc=`echo The number of users is `who| wc-l``

$ echo $abc

The number of users is 5

$

在反引号之间的命令行中也可以使用shell的特殊字符。Shell为得到``中命令的结果，它实际上要去执行``中指定的命令。执行时，命令中的特殊字符，如$，”，?等又将具有特殊含义，并且``所包含的可以是任何一个合法的Shell命令，如：

$ ls

note readme.txt Notice Unix.dir

$ TestString＝”`echo $HOME ` ` ls ［nN］*`”

$ echo $TestString

/home/yxz note Notice

$

其他情况，读者可自行试之。

1. 注释符

在shell编程中经常要对某些正文行进行注释，以增加程序的可读性。在Shell中以字符“#”开头的正文

行表示注释行。

此外还有一些特殊字符如：用于输入/输出重定向与管道的<、>、<<、>>和|；执行后台命令的&；命令执行操作符&&和||及表示命令组的{}将在下面各小节中加以介绍。
标准输入/输出和重定向

1. 标准输入与输出

我们知道，执行一个shell命令行时通常会自动打开三个标准文件，即标准输入文件（stdin），通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），这两个文件都对应终端的屏幕。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

我们以cat命令为例，cat命令的功能是从命令行给出的文件中读取数据，并将这些数据直接送到标准输出。若使用如下命令：

$ cat config

将会把文件config的内容依次显示到屏幕上。但是，如果cat的命令行中没有参数，它就会从标准输入中读取数据，并将其送到标准输出。例如：

$ cat

Hello world

Hello world

Bye

Bye

<ctrl+d>

$

用户输入的每一行都立刻被cat命令输出到屏幕上。

另一个例子，命令sort按行读入文件正文（当命令行中没有给出文件名时，表示从标准输入读入），将其排序，并将结果送到标准输出。下面的例子是从标准输入读入一个采购单，并将其排序。

$ sort

bananas

carrots

apples

<ctrl+d>

apples

bananas

carrots

$

这时我们在屏幕上得到了已排序的采购单。

直接使用标准输入/输出文件存在以下问题：

输入数据从终端输入时，用户费了半天劲输入的数据只能用一次。下次再想用这些数据时就得重新输入。而且在终端上输入时，若输入有误修改起来不是很方便。

输出到终端屏幕上的信息只能看不能动。我们无法对此输出作更多处理，如将输出作为另一命令的输入进行进一步的处理等。

为了解决上述问题，Linux系统为输入、输出的传送引入了另外两种机制，即输入/输出重定向和管道。

2. 输入重定向

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以不来自键盘，而来自一个指定的文件。所以说，输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

例如，命令wc统计指定文件包含的行数、单词数和字符数。如果仅在命令行上键入：

$ wc

wc将等待用户告诉它统计什么，这时shell就好象死了一样，从键盘键入的所有文本都出现在屏幕上，但并没有什么结果，直至按下＜ctrl+d＞，wc才将命令结果写在屏幕上。

如果给出一个文件名作为wc命令的参数，如下例所示，wc将返回该文件所包含的行数、单词数和字符数。

$ wc /etc/passwd

20 23 726 /etc/passwd

$

另一种把/etc/passwd文件内容传给wc命令的方法是重定向wc的输入。输入重定向的一般形式为：命令<文件名。可以用下面的命令把wc命令的输入重定向为/etc/passwd文件：

```
$ wc < /etc/passwd

20 23 726

$
```

另一种输入重定向称为here文档，它告诉shell当前命令的标准输入来自命令行。here文档的重定向操作符使用<<。它将一对分隔符（本例中用delim表示）之间的正文重定向输入给命令。下例将一对分隔符delim之间的正文作为wc命令的输入，统计出正文的行数、单词数和字符数。

```
$ wc<<delim

>this text forms the content

>of the here document,which

>continues until the end of

>text delimter

>delim

4 17 98
```

在<<操作符后面,任何字符都可以作为正文开始前的分隔符，本例中使用delim作为分隔符。here文档的正文一直延续到遇见另一个分隔符为止。第二个分隔符应出现在新行的开头。这时here文档的正文（不包括开始和结束的分隔符）将重新定向送给命令wc作为它的标准输入。

由于大多数命令都以参数的形式在命令行上指定输入文件的文件名，所以输入重定向并不经常使用。尽管如此，当要使用一个不接受文件名作为输入参数的命令，而需要的输入内容又存在一个文件里时，就能用输入重定向解决问题。

1. 输出重定向

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

输出重定向比输入重定向更常用，很多情况下都可以使用这种功能。例如，如果某个命令的输出很多，在屏幕上不能完全显示，那么将输出重定向到一个文件中，然后再用文本编辑器打开这个文件，就可以查看输出信息；如果想保存一个命令的输出，也可以使用这种方法。还有，输出重定向可以用于把一个命令的输出当作另一个命令的输入（还有一种更简单的方法，就是使用管道，将在下面介绍）。

输出重定向的一般形式为：命令>文件名。例如：

```
$ ls > directory.out
```

$ cat directory.out

ch1.doc ch2.doc ch3.doc chimp config mail/ test/

$

将ls命令的输出保存为一个名为directory.out的文件。

注：如果>符号后边的文件已存在，那么这个文件将被重写。

为避免输出重定向中指定文件只能存放当前命令的输出重定向的内容，shell提供了输出重定向的一种追加手段。输出追加重定向与输出重定向的功能非常相似，区别仅在于输出追加重定向的功能是把命令（或可执行程序）的输出结果追加到指定文件的最后，而该文件原有内容不被破坏。

如果要将一条命令的输出结果追加到指定文件的后面，可以使用追加重定向操作符>>。形式为：命令>>文件名。例如：

$ ls *.doc>>directory.out

$ cat directory.out

ch1.doc ch2.doc ch3.doc chimp config mail/ test/

ch1.doc ch2.doc ch3.doc

$

和程序的标准输出重定向一样，程序的错误输出也可以重新定向。使用符号2>（或追加符号2>>）表示对错误输出设备重定向。例如下面的命令：

$ ls /usr/tmp 2> err.file

可在屏幕上看到程序的正常输出结果，但又将程序的任何错误信息送到文件err.file中，以备将来检查用。

还可以使用另一个输出重定向操作符（&>）将标准输出和错误输出同时送到同一文件中。例如：

$ ls /usr/tmp &> output.file

利用重定向将命令组合在一起，可实现系统单个命令不能提供的新功能。例如使用下面的命令序列：

$ ls /usr/bin > /tmp/dir

$ wc –w < /tmp/dir

统计了/usr/bin目录下的文件个数。

管 道

将一个程序或命令的输出作为另一个程序或命令的输入，有两种方法，一种是通过一个临时文件将两个命令或程序结合在一起，例如上个例子中的/tmp/dir文件将ls和wc命令联在一起；另一种是Linux所提供的管道功能。这种方法比前一种方法更好。

管道可以把一系列命令连接起来，这意味着第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入，以此类推。显示在屏幕上的是管道行中最后一个命令的输出（如果命令行中未使用输出重定向）。

通过使用管道符“|”来建立一个管道行。用管道重写上面的例子：

$ ls /usr/bin|wc -w

1789

再如：

$ cat sample.txt|grep "High"|wc -l

管道将cat命令（列出一个文件的内容）的输出送给grep命令。grep命令在输入里查找单词High，grep命令的输出则是所有包含单词High的行，这个输出又被送给wc命令，wc命令统计出输入中的行数。假设sample.txt文件的内容如下：

Things to do today:

Low:Go grocery shopping

High:Return movie

High:Clear level 3 in Alien vs. Predator

Medium:Pick up clothes from dry cleaner

那么该管道行的结果是2。

命令替换

命令替换和重定向有些相似，但区别在于命令替换是将一个命令的输出作为另外一个命令的参数。常用命令格式为：

command1 `command2`

其中，command2的输出将作为command1的参数。需要注意的是这里的`符号，被它括起来的内容将作为命令执行，执行后的结果作为command1的参数。例如：

$ cd `pwd`

该命令将pwd命令列出的目录作为cd命令的参数，结果仍然是停留在当前目录下。

第二十二课　在Bash中的操作　　　　　2000年/5月/29日

命令和文件名扩展特性

Bash命令行具有命令和文件名扩展特性。当输入一个还没完成的命令或文件名时，只需键入Tab键就能激活命令和文件名扩展特性，从而完成该命令的剩余输入。如果有多个命令或文件的前缀相同，Bash将响铃并等待用户输入足够的字符，以便选择唯一的命令或文件名，如果找到，系统将自动补齐搜索到的命令或文件名，用户按回车键后，系统将执行这条指令。例如：

$ cat pre <Tab>

$ cat preface

Bash也能列出当前目录下部分匹配的文件名来完成文件名扩展。如果键入Esc，然后键入?，shell将列出所有与输入的字符串相匹配的文件名。例如下例，在没有完成的输入后键入Esc ?，shell将列出所有与输入的字符串相匹配的字符串，然后shell回显命令行，根据列出的文件名，可以键入要输入的文件名或按下Tab键来完成文件名扩展。例如：

$ ls

document docudrama

$ cat doc <ESC ?>

document

docudrama

$ cat docudrama

［例】下面是一个目录包含的文件列表：

Firebird2.7.tgz Firebird.README Firebird2.60.tgz

FireBird Firebird2.60.tgz.README

现在要删除Firebird2.60.tgz.README文件，键入：

$ rm －f Fi<Tab>

系统会发出警报声，并且自动将命令行补全为：

$ rm －f Fire

并等待用户进一步输入文件名的后面部分。现在再键入：

b<Tab>

系统再次发出警报声，并且自动将命令行补全为：

$ rm －f Firebird

并等待用户进一步输入文件名的后面部分。现在再键入：

2.6<Tab>

系统再次发出警报声，并且自动将命令行补全为：

$ rm －f Firebird2.60.tgz

并等待用户进一步输入文件名的后面部分。现在再键入：

.<Tab>

此时命令将被补全为：

$ rm －f Firebird2.60.tgz..README

从上例可以看到，bash总是尽力根据用户输入的信息来补全命令。当无法根据现有信息补全命令时，则提示用户再给出更多的信息，然后再根据用户的提示来进一步补全命令。作为用户最好是能够一次性给出足够的信息以便于bash进行命令补全；否则多按几次<Tab>，时间也就消耗掉了。

命令行编辑

在Bash中可以对命令行进行编辑，以便用户在执行所键入的命令之前能够修改所键入的命令。如果在键入命令时出现拼写错误，只需在运行所键入的命令之前，使用编辑命令来纠正编辑错误，然后执行它，而不用重新输入整行命令。这个功能对以长路径文件名作参数的命令特别有用。

表10-2是对命令行编辑操作的一个总结。

表10-2　命令行编辑操作

命令行编辑操作

功能

Ctrl+b或左箭头键

左移一个字符（移至前一个字符）

Ctrl+f或右箭头键

右移一个字符（移至后一个字符）

Ctrl+a

移至行首

Ctrl+e

移至行尾

Esc b

左移一个单词

Esc f

右移一个单词

Del

删除光标所在处的字符

Ctrl+d

删除光标所在处的字符

BACKSPACE或Ctrl+h

删除光标左边的字符

Ctrl+k

删除至行尾

命令历史

在Bash中，history命令能够保存最近所执行的命令。这些命令的历史记录号从1开始，只有有限个命令可以被保存起来，最多500个，即history命令的历史记录号缺省值为500。要查看最近执行的命令，只要键入history命令，然后键入回车键，最近执行过的命令即按先后顺序被显示出来（各条命令前的数字为历史记录号）。

［例】

$ history

1 cp mydata today

2 vi mydata

3 mv mydata reports

4 cd reports

5 ls

…

所有这些命令都被称为事件（event），一个事件表示一个操作已经发生，即一个命令已被执行。这些事件根据它们被执行的先后顺序用数字标识，这一标识称为历史事件号。最后执行的历史事件的事件号最大。每个事件都可由它的历史事件号或命令的初始字符或字符串等确定。

利用history命令能够查询以前的事件，并可把它们显示到命令行上执行这一事件。最简便的方法就是利用上下箭头键，把先前的事件逐次显示到命令行。这个操作不需要运行history命令就可以执行。按动一下上箭头键，那么上一次执行的一个事件就将出现在命令行上，再按一下，上一次的前一事件又会出现在命令行上；按动一下下箭头键，将会使当前事件的下一事件出现在命令行上。

Bash也可以通过键入Esc、Tab键来完成对历史事件的字符扩展。和标准命令行扩展特性一样，键入历史事件的部分字符串，然后键入Esc，再键入Tab键，与刚才键入的字符串相匹配的历史事件将自动扩展并回显到命令行处。如果不止一个事件与输入的字符串相匹配，就会听到一声响铃，继续键入字符或字符串，shell将会唯一确定用户所要键入的历史事件。

还有一个查询和执行历史事件的命令——!命令。在!命令后键入与历史事件相关联的字符，这个关联字符可以是历史事件的历史事件号，也可以是该事件的前几个字符。在下面的例子中，查询到历史事件号为3的事件，然后又用其开头的几个字符去匹配，也查询到该命令。

［例】

$!3

mv mydata reports

$!mv

mv mydata reports

也可以用一个偏移量（相对于历史事件列表中最后一个事件）来查询历史事件。负的偏移量将从历史事件列表表尾向前偏移。在下面的例子中，历史事件号为2的事件“vi mydata”就是用一个负的偏移量查

询到的。必须注意的是，这个偏移量是相对于历史事件列表中的最后一个事件的。在本例中，历史事件列表中最后一个事件是事件5，历史事件列表中第一个事件为1。从历史事件号为5的事件，往前偏移4，即是历史事件号为2的事件。

［例】

$!-4

vi mydata

如果键入!!，则系统默认为上一事件。下面的例子中，用户在命令行上键入!!命令，系统将执行上一事件：“ls”命令。

［例】

$!!

ls

mydata today reports

也可以用“模式”来搜索一个历史事件。搜索的“模式”必须用符号“?”括起来。下例是用“模式”“?myd?”来搜索历史事件号为3的历史事件“vi mydata”。

［例】

$!?myd?

vi mydata

1. 查询历史事件

可以在命令行上编辑历史事件列表中的事件。表10-3列出了查询历史事件列表的各种操作。

表10-3　查询历史事件操作

查询历史事件操作

功能

Ctrl+n或向下光标键

移至历史事件列表中当前事件的下一历史事件

Ctrl+p或向上光标键

移至历史事件列表中当前事件的前一历史事件

Esc <

移至历史事件列表表首

Esc >

移至历史事件列表表尾

!event_num

用历史事件号来定位一个历史事件

!characters

用历史事件的字符前缀来查询一个历史事件

!?pattern

用“模式”来查询历史事件列表中的事件

!-event_num

通过偏移量来定位历史事件
2. 配置history：HISTFILE及HISTSIZE

系统保存的历史事件数被保存在一个特定的系统变量中，这个变量就是HISTSIZE。这个变量的缺省值通常被设置为500。这个值可以被修改。例如：

$ HISTSIZE=10

将HISTSIZE的值重新设置为10。

历史事件被保存在一个文件中，文件名由变量HISTFILE指定。通常这个文件的缺省名是.bash_history。通过给变量HISTFILE赋值，可以指定新的文件名。

［例】

$ echo $HISTFILE

/home/lisa/.bash_history

$ HISTFILE=”/home/lisa/newhist”

$ echo $HISTFILE

/home/lisa/newhist

以上操作先显示变量HISTFILE的值，然后赋予它新的值“/home/lisa/newhist”，以后所有的历史事件将被保存在newhist文件中。

____
碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:15am 发表主题:

别名

还有一个使工作变得轻松的方法是使用命令别名。命令别名通常是其他命令的缩写，用来减少键盘输入。

命令格式为:

alias [alias-name=’original-command’]

其中，alias-name是用户给命令取的别名，original-command是原来的命令和参数。需要注意的是，由于Bash是以空格或者回车来识别原来的命令的，所以如果不使用引号就可能导致Bash只截取第一个字，从而出现错误。如果alias命令后面不使用任何参数，则显示当前正在使用的被别名化的命令及其别名。为命令取的别名在该次登录期间始终有效。如果用户需要别名在每次登录时都有效，那么就将alias命令写到初始化脚本文件中。

[例]如果经常要键入如下的命令，最好为它建立一个别名来减少工作量。

$ cd /usr/X11/lib/X11

假如为这个长命令建立一个名为goconfig的别名，在Bash提示符下键入如下命令：

$ alias goconfig=’cd /usr/X11/lib/X11’

现在，除非您退出Bash，键入goconfig将和原来的长命令有同样的作用。如果想取消别名，可以使用下面的命令：

$ unalias goconfig

这是一些很多人认为有用的别名，可以把它们写入初始化脚本文件中来提高工作效率：

alias ll=’ls –l’

alias log=’logout’

alias ls=’ls –F’

如果您是一名DOS用户并且习惯了DOS命令，可以用下面的别名定义使Linux表现得象DOS一样：

alias dir=’ls’

alias copy=’cp’

alias rename=’mv’

alias md=’mkdir’

alias rd=’rmdir’

注意：在定义别名时，等号两边不能有空格，否则shell不能决定您需要做什么。仅在命令中包含空格或特殊字符时才需要引号。

如果键入不带任何参数的alias命令，将显示所有已定义的别名。

提示符

Bash有两级提示符。第一级提示符是经常见到的Bash在等待命令输入时的情况。第一级提示符的默认值是$符号。如果用户不喜欢这个符号，或者愿意自己定义提示符，只需修改PS1变量的值。例如将其改为：

PS1=”Enter a command:”

第二级提示符是当Bash为执行某条命令需要用户输入更多信息时显示的。第二级提示符默认为>。如果需要自己定义该提示符，只需改变PS2变量的值。例如将其改为：

PS2=”More information:”

上面的两个例子都是设定提示符为静态字符串的情况。其实用户也可以使用一些事先已经定义好的特殊

字符。这些特殊字符将使提示符中包含当前时间之类的信息。表10-4列出了最常用的一些特殊字符及其含义。

表10-4　bash提示符常用特殊字符

特殊字符	说 明
!	显示该命令的历史编号
#	显示shell激活后，当前命令的历史编号
$	显示一个$符号，如果当前用户是root则显示#符号
\	显示一个反斜杠
d	显示当前日期
h	显示运行该shell的计算机主机名
n	打印一个换行符，这将导致提示符跨行
s	显示正在运行的Shell的名称
t	显示当前时间

u

显示当前用户的用户名

W

显示当前工作目录基准名

w

显示当前工作目录

这些特殊字符可以组合起来，为用户提供一些提示符，提供很有用的信息。下面来看几个实际例子：

PS1=”t”

将使提示符变成如下所示：

02:16:15

而 PS1=t

将使提示符变成如下所示：

t

若PS1=”t\”

将使提示符变成如下所示：

02:16:30

该例就是使用两个特殊字符的组合得到的。

控制shell的运行方式

Bash有一些特殊变量，能控制shell以不同的方式工作。例如，变量noclobber能防止在重定向输出时意外地覆盖一个文件。通过set命令可以设置noclobber变量的有效或无效。set命令有两个参数：一个是指定变量开（on）或关（off）的选项，一个是特殊变量的变量名。要使某一特殊变量开（有效），用-o选项，要使其关（无效），用+o选项。例如：

$ set －o noclobber // 使noclobber变量开

$ set +o noclobber // 使noclobber变量关

三个最常用的shell特殊变量有：ignoreeof、noclobber及noglob。

ignoreeof

ignoreeof变量用来禁止使用ctrl+d来退出shell（ctrl+d不仅用来退出shell，而且可以终止用户直接输往标准输出上的输入。该操作经常在一些shell实用命令中使用，例如实用命令cat。在这些实用程序操作中，非常容易误操作而意外地退出shell。ignoreeof特殊变量正是用来防止这种意外的退出。例如：

$ set －o ignoreeof

之后，用户只能用logout或exit命令退出shell。

noclobber

noclobber变量可以在重定向输出时保护已存在的文件，防止被意外地覆盖。在下例中，用户设置noclobber为有效，在重定向时，用户试图去覆盖已经存在的文件myfile，此时系统将返回一个错误信息。

［例］

$ set －o noclobber

$ cat preface>myfile

bash: myfile: cannot overwrite existing file

$

noglob

设置noglob变量后，shell将不扩展文件名中一些特殊的字符或字符串。如字符*、?、［ ］等将不再作为通配符。如果用户希望列出结尾为?的文件名answer?，可通过如下步骤：首先，用户使noglob变量为无效，然后再列出文件名。可以看到，目前命令行上的问号?被认为是文件名中的一个字符，而不再被看作通配符。

$ set －o noglob

$ ls answer?

answer?

子shell与export命令

用户登录到Linux系统后，系统将启动一个用户shell。在这个shell中，可以使用shell命令或声明变量，也可以创建并运行shell脚本程序。运行shell脚本程序时，系统将创建一个子shell。此时，系统中将有两个shell，一个是登录时系统启动的shell，另一个是系统为运行脚本程序创建的shell。当一个脚本程序运行完毕，它的脚本shell将终止，可以返回到执行该脚本之前的shell。从这种意义上来说，用

户可以有许多shell，每个shell都是由某个shell（称为父shell）派生的。

在子shell中定义的变量只在该子shell内有效。如果在一个shell脚本程序中定义了一个变量，当该脚本程序运行时，这个定义的变量只是该脚本程序内的一个局部变量，其他的shell不能引用它，要使某个变量的值可以在其他shell中被改变，可以使用export命令对已定义的变量进行输出。export命令将使系统在创建每一个新的shell时定义这个变量的一个拷贝。这个过程称之为变量输出。

［例］在本例中，变量myfile是在dispfile脚本程序中定义的。然后用export命令将变量myfile输出至任何子shell，例如当执行printfile脚本程序时产生的子shell。

dispfile脚本程序清单：

/**************begin dispfile**************/

myfile=”List”

export myfile

echo “Displaying $myfile”

pr －t －n $myfile

printfile

/**************end dispfile***************/

printfile脚本程序清单：

/**************begin printfile**************/

echo “Printing $myfile”

lpr $myfile&

/**************end printfile**************/

$dispfile

Displaying List

1 screen

2 modem

3 paper

Printing List

$

定制Bash

在本节中已经介绍了很多定制Bash的方法，但是迄今为止，这些方法都只是对当前Bash对话有用。只要用户退出登录，所做的一切改变都会丢失。所以应该在Bash的初始化文件中做永久性的修改。

用户可以将每次启动Bash所需要执行的命令放入初始化文件中，最常见的命令就是alias命令和变量定义两种。系统中的每个用户在其主目录中都有一个.bash_profile文件，Bash每次启动时都将读取该文件，其中包含的所有命令都将被执行。

下面便是默认.bash_profile文件的代码：

```
#.bash_profile

#Get the aliases and functions

if  [-f ~/.bashrc ] ;then

.~/.bashrc

fi

#User specific environment and startup programs

PATH=$PATH:$HOME/bin

ENV=$HOME/.bashrc

USERNAME=” ”

Export USERNAME ENV PATH
```

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:16am 发表主题:

网络相关ping finger nslookup netstat[转帖]
netstat命令

netstat命令的功能是显示网络连接、路由表和网络接口信息，可以让用户得知目前都有哪些网络连接正在运作。

该命令的一般格式为：

netstat ［选项］

命令中各选项的含义如下：

-a 显示所有socket，包括正在监听的。

-c 每隔1秒就重新显示一遍，直到用户中断它。

-i 显示所有网络接口的信息，格式同“ifconfig -e”。

-n 以网络IP地址代替名称，显示出网络连接情形。

-r 显示核心路由表，格式同“route -e”。

-t 显示TCP协议的连接情况。

-u 显示UDP协议的连接情况。

-v 显示正在进行的工作。

［例］在本地机上使用netstat命令。

$ netstat

Active Internet connections （w/o servers）

Proto Recv-Q Send-Q Local Address Foreign Address State

Active UNIX domain sockets （w/o servers）

Proto RefCnt Flags Type State I-Node Path

unix 1 ［ ］ STREAM CONNECTED 270 @00000008

unix 1 ［ ］ STREAM CONNECTED 150 @00000002

unix 1 ［ ］ STREAM CONNECTED 104 @00000001

unix 1 ［ ］ STREAM CONNECTED 222 @00000004

unix 1 ［ ］ STREAM CONNECTED 171 @00000003

unix 1 ［ ］ STREAM CONNECTED 271 /dev/log

unix 1 ［ ］ STREAM CONNECTED 225 /dev/log

unix 1 ［ ］ STREAM CONNECTED 223 /dev/log

unix 1 ［ ］ STREAM CONNECTED 203 /dev/log

unix 1 ［ ］ STREAM CONNECTED 105 /dev/log

……

nslookup命令

nslookup命令的功能是查询一台机器的IP地址和其对应的域名。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器，就可以用这个命令查看不同主机的IP地址对应的域名。

该命令的一般格式为：

nslookup ［IP地址/域名］

［例］在本地机上使用nslookup命令。

$ nslookup

Default Server: name.tlc.com.cn

Address: 192.168.1.99

>

在符号“>”后面输入要查询的IP地址或域名并回车即可。如果要退出该命令，输入exit并回车即可。

finger命令

finger命令的功能是查询用户的信息，通常会显示系统中某个用户的用户名、主目录、停滞时间、登录时间、登录shell等信息。如果要查询远程机上的用户信息，需要在用户名后面接“@主机名”，采用［用户名@主机名］的格式，不过要查询的网络主机需要运行finger守护进程。

该命令的一般格式为：

finger ［选项］ ［使用者］ ［用户@主机］

命令中各选项的含义如下：

-s 显示用户的注册名、实际姓名、终端名称、写状态、停滞时间、登录时间等信息。

-l 除了用-s选项显示的信息外，还显示用户主目录、登录shell、邮件状态等信息，以及用户主目录下的.plan、.project和.forward文件的内容。

-p 除了不显示.plan文件和.project文件以外，与-l选项相同。

［例］在本地机上使用finger命令。

$ finger xxq

Login: xxq Name:

Directory: /home/xxq Shell: /bin/bash

Last login Thu Jan 1 21:43 （CST） on tty1

No mail.

No Plan.

$ finger

Login Name Tty Idle Login Time Office Office Phone

root root *1 28 Nov 25 09:17

……

ping命令

ping命令用于查看网络上的主机是否在工作，它向该主机发送ICMP ECHO_REQUEST包。有时我们想从网络上的某台主机上下载文件，可是又不知道那台主机是否开着，就需要使用ping命令查看。

该命令的一般格式为：

ping ［选项］ 主机名/IP地址

命令中各选项的含义如下：

-c 数目 在发送指定数目的包后停止。

-d 设定SO_DEBUG的选项。

-f 大量且快速地送网络封包给一台机器，看它的回应。

-I 秒数 设定间隔几秒送一个网络封包给一台机器，预设值是一秒送一次。

-l 次数 在指定次数内，以最快的方式送封包数据到指定机器（只有超级用户可以使用此选项）。

-q 不显示任何传送封包的信息，只显示最后的结果。

-r 不经由网关而直接送封包到一台机器，通常是查看本机的网络接口是否有问题。

-s 字节数 指定发送的数据字节数，预设值是56，加上8字节的ICMP头，一共是64ICMP数据字节。[/b]

____
碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:17am 发表主题:

Linux下的打印工作命令[转帖]
在Linux下采用假脱机（spooling）打印方法，当用户需要打印一个文件时，该文件并不直接送到打印机，而是送到spool目录下，然后由一个负责打印的后台进程把这些数据送入打印机。

Linux对每台打印机都定义了一个打印缓冲区，打印机守护程序经常扫描打印缓冲区以查看有无要打印的新文件。如果存在，就按先进先出的顺序打印缓冲区中的文件。

Linux系统除了可以在本地打印机上打印外，还可以通过网络打印机远程打印。

用于打印的命令

Linux系统提供了一组有关打印的命令。一般情况下，打印命令使用默认打印机；如果用户定义了PRINTER环境变量，打印命令就使用这个变量定义的打印机；另外，用户还可以在命令行上指定要使用的打印机。

lpr命令

lpr命令是脱机打印命令，该命令将打印作业放到打印缓冲队列中。为Linux系统指定的每台打印机都有自己的打印缓冲目录，每个目录中的minfree文件指定保存打印文件的磁盘块的数量。

lpr同lpd守护进程通讯，lpd扫描/etc/printcap文件，查询打印机对应的缓存目录，然后由lpd控制打印，将需要打印的数据送到实际打印机上。如果没有指定文件，lpr就使用标准输入。

lpr命令的格式为：

lpr ［-P printer］ ［#num］ ［其他选项］ ［name…］

命令中各选项的含义如下：

-P 指定打印机。如果不用此选项，则使用缺省打印机或环境变量PRINTER指定的打印机。

-m 打印完毕后发送email。

-#num 打印num份。

lpq命令

lpq是缓冲队列检查命令，它通过lpd在缓冲区中检查打印文件，报告指定作业的状态或指定用户的所有作业。不带任何参数的lpq命令显示现在队列中的任何作业。lpq命令的显示结果中一个重要的信息就是作业标识号（作业ID），它标识一个特定的作业。如果用户想取消一个挂起的作业，就必须在命令中指定这个标识号。

lpq命令的格式为：

lpq ［-l］ ［-P printer］ ［job #］ ［user…］

命令中各选项的含义如下：

-P 指定一个打印机，否则使用缺省打印机或环境变量PRINTER指定的打印机。

-l 打印组成作业的所有文件的信息。

对提交的每一个作业，lpq报告用户名、在队列中的级别、组成作业的文件、作业标识以及总的大小等信息。

lprm命令

lprm命令用于从缓冲队列中删除打印作业，用户可以使用该命令从缓冲队列中删除属于自己的一个或多个打印作业。

lprm命令的格式为：

lprm ［-P printer］ ［-］ ［job #］ ［user…］

命令中各选项的含义如下：

-P 指定一个打印机，否则使用缺省打印机或环境变量PRINTER指定的打印机。

- 删除用户所有的打印作业。

user 删除队列中属于用户user的作业（只有超级用户可以这样做）。

job # 通过指定作业号#删除某个打印作业，作业号可以通过lpq命令得到，如：

$ lpq －l

lst:ken ［job #013ucbarpa］

（standard input） 100 bytes

$ lprm 13

打印缓冲区目录

对打印而言，有一个非常重要的目录，就是打印缓冲区目录，要打印的数据在被打印之前都集中到这里。通常一台打印机对应一个打印缓冲区目录，这样比较容易管理打印机。例如，系统使用/var/spool/lpd作为主打印缓冲区，每个单独的打印机都在主打印缓冲区下有一个与这台打印机同名的目录。因此，名为ps_nff的打印机把/var/spool/lpd/ps_nff作为它的打印缓冲区目录。

碧轩
人之初，性本色！
http://www.ourlinux.net/

http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:18am 发表主题:

与系统管理有关的命令
wall命令

这个命令的功能是对全部已登录的用户发送信息，用户可以先把要发送的信息写好存入一个文件中，然后输入：

wall < 文件名

这样就能对所有的用户发送信息了。

在上面的例子中符号“<”表示输入重定向，有关它的含义和用法请参阅第十章的有关内容。

例如：

wall ‘Thank you!’

Broadcast message from root （tty1） Fri Nov 26 14：15：07 1999…

Thank you!

#

执行以上命令后，用户的屏幕上显示出“Thank you!”信息后，并不出现系统提示符$（#），再次按回车键后，屏幕出现系统提示符。

write命令

write命令的功能是向系统中某一个用户发送信息。

该命令的一般格式为：

write 用户帐号 [终端名称]

例如：

$ write xxq

hello

此时系统进入发送信息状态，用户可以输入要发送的信息，输入完毕，希望退出发送状态时，按组合键即可。

上述命令执行的结果是，用户xxq的屏幕上会显示：

message from test@test.tlc.com.cn tty1 at 15:51…

hello

EOF

mesg指令

mesg命令设定是否允许其他用户用write命令给自己发送信息。如果允许别人给自己发送信息，输入命令：

mesg y

否则，输入：

mesg n

对于超级用户，系统的默认值为 n；而对于一般用户系统的默认值为y。

如果mesg后不带任何参数，则显示当前的状态是y还是n，如：

$ mesg

is y

或：

mesg

is n

sync命令

sync命令是在关闭Linux系统时使用的。

用户需要注意的是，不能用简单的关闭电源的方法关闭系统，因为Linux象其他Unix系统一样，在内存中缓存了许多数据，在关闭系统时需要进行内存数据与硬盘数据的同步校验，保证硬盘数据在关闭系统时是最新的，只有这样才能确保数据不会丢失。一般正常的关闭系统的过程是自动进行这些工作的，在系统运行过程中也会定时做这些工作，不需要用户干预。

sync命令是强制把内存中的数据写回硬盘，以免数据的丢失。用户可以在需要的时候使用此命令。该命令的一般格式为：

sync

shutdown命令

shutdown 命令可以安全地关闭或重启Linux系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。精确时间的格式是hh:mm，表示小时和分钟；时间段由“+”和分钟数表示。系统执行该命令后，会自动进行数据同步的工作。该命令的一般格式为：

shutdown ［选项］ ［时间］ ［警告信息］

命令中各选项的含义为：

- k 并不真正关机，而只是发出警告信息给所有用户。

- r 关机后立即重新启动。

- h 关机后不重新启动。

- f 快速关机，重启动时跳过fsck。

- n 快速关机，不经过init程序。

- c 取消一个已经运行的shutdown。

需要特别说明的是，该命令只能由超级用户使用。

例1：系统在十分钟后关机，并且马上重新启动。

shutdown – r +10

例2：系统马上关机，并且不重新启动。

shutdown – h now

free命令

free命令的功能是查看当前系统内存的使用情况，它显示系统中剩余及已用的物理内存和交换内存，以及共享内存和被核心使用的缓冲区。该命令的一般格式为：

free [-b | -k | -m]

命令中各选项的含义如下：

-b 以字节为单位显示。

-k 以K字节为单位显示。

-m 以兆字节为单位显示。

例：

$ free

total used free shared buffers cached

Mem: 63076 32020 31056 8204 16360 6048

-/+ buffers/cache: 9612 53464

Swap: 64476 2240 62236

uptime命令

uptime命令显示系统已经运行了多长时间，它依次显示下列信息：现在时间、系统已经运行了多长时间、目前有多少登录用户、系统在过去的1分钟、5分钟和15分钟内的平均负载。该命令的一般格式为：

uptime

例：

uptime

4:43pm up 1 day, 5:51, 2 user, load average: 0.01, 0.01, 0.00

与用户有关的命令

passwd命令

出于系统安全考虑，Linux系统中的每一个用户除了有其用户名外，还有其对应的用户口令。因此使用useradd命令增加时，还需使用passwd命令为每一位新增加的用户设置口令；用户以后还可以随时用passwd命令改变自己的口令。该命令的一般格式为：

passwd ［用户名］

其中用户名为需要修改口令的用户名。只有超级用户可以使用“passwd 用户名”修改其他用户的口令，普通用户只能用不带参数的passwd命令修改自己的口令。该命令的使用方法如下：

输入passwd< Enter>;

在（current） UNIX passwd:下输入当前的口令

在new password:提示下输入新的口令（在屏幕上看不到这个口令）：

系统提示再次输入这个新口令。

输入正确后，这个新口令被加密并放入/etc/shdow文件。选取一个不易被破译的口令是很重要的。选取口令应遵守如下规则：

口令应该至少有六位（最好是八位）字符；

口令应该是大小写字母、标点符号和数字混杂的。

超级用户修改其他用户（xxq）的口令的过程如下，

passwd xxq

New UNIX password:

Retype new UNIX password:

passwd: all authentication tokens updated successfully

#

su命令

这个命令非常重要。它可以让一个普通用户拥有超级用户或其他用户的权限，也可以让超级用户以普通用户的身份做一些事情。普通用户使用这个命令时必须有超级用户或其他用户的口令。如要离开当前用户的身份，可以打exit。

该命令的一般形式为：

su ［选项］ ［? ］ ［使用者帐号］

说明：若没有指定使用者帐号，则系统预设值为超级用户root。

该命令中各选项的含义分别为：

? c 执行一个命令后就结束。

? 加了这个减号的目的是使环境变量和欲转换的用户相同。

? m 保留环境变量不变。

例1： 变成root用户

$ su ?

password: 【输入超级用户的密码】

例2： 变成xu使用者，并执行一个命令就结束。

$ su -xu ? c “rmdir cat1”

其他命令

echo命令

echo命令的功能是在显示器上显示一段文字，一般起到一个提示的作用。该命令的一般格式为：

echo ［ -n ］ 字符串

其中选项n表示输出文字后不换行；字符串可以加引号，也可以不加引号。用echo命令输出加引号的字符串时，将字符串原样输出；用echo命令输出不加引号的字符串时，将字符串中的各个单词作为字符串输出，各字符串之间用一个空格分割。

例1：

$ echo ‘Thank you !’

Thank you !

例2：

$ echo Thank you !

Thank you !

cal命令

cal命令的功能是显示某年某月的日历。该命令的一般格式为：

cal ［选项］ ［月 ［年］］

命令中各选项的含义为：

- j 显示出给定月中的每一天是一年中的第几天（从1月1日算起）。

- y 显示出整年的日历。

例1 显示1999年11月的日历。

$ cal 11 1999

November 1999

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30

例2 显示1999年11月的每一天是一年中的第几天。

$ cal – j 11 1999

November 1999

Sun Mon Tue Wed Thu Fri Sat

305 306 307 308 309 310

311 312 313 314 315 316 317

318 319 320 321 322 323 324

325 326 327 328 329 330 331

332 333 334

date命令

date命令的功能是显示和设置系统日期和时间。该命令的一般格式为：

date ［选项］ 显示时间格式（以+开头，后面接格式）

date 设置时间格式

命令中各选项的含义分别为：

-d datestr, --date datestr 显示由datestr描述的日期

-s datestr, --set datestr 设置datestr 描述的日期

-u, --universal 显示或设置通用时间

时间域

% H 小时（00..23）

% I 小时（01..12）

% k 小时（0..23）

% l 小时（1..12）

% M 分（00..59）

% p 显示出AM或PM

% r 时间（hh：mm：ss AM或PM），12小时

% s 从1970年1月1日00：00：00到目前经历的秒数

% S 秒（00..59）

% T 时间（24小时制）（hh:mm:ss）

% X 显示时间的格式（％H:％M:％S）

% Z 时区

日期域

% a 星期几的简称（ Sun..Sat）

% A 星期几的全称（ Sunday..Saturday）

% b 月的简称（Jan..Dec）

% B 月的全称（January..December）

% c 日期和时间（ Mon Nov 8 14：12：46 CST 1999）

% d 一个月的第几天（01..31）

% D 日期（mm / dd / yy）

% h 和%b选项相同

% j 一年的第几天（001..366）

% m 月（01..12）

% w 一个星期的第几天（0代表星期天）

% W 一年的第几个星期（00..53，星期一为第一天）

% x 显示日期的格式（mm/dd/yy）

% y 年的最后两个数字（ 1999则是99）

% Y 年（例如：1970，1996等）

需要特别说明的是，只有超级用户才能用date命令设置时间，一般用户只能用date命令显示时间。

例1：用指定的格式显示时间。

$ date ‘+This date now is =>%x , time is now =>%X , thank you !’

This date now is =>11/12/99 , time is now =>17:53:01 , thank you !

例2：用预定的格式显示当前的时间。

date

Fri Nov 26 15：20：18 CST 1999

例3：设置时间为下午14点36分。

date -s 14:36:00

Fri Nov 26 14：15：00 CST 1999

例4：设置时间为1999年11月28号。

date -s 991128

Sun Nov 28 00：00：00 CST 1999

clear命令

clear命令的功能是清除屏幕上的信息，它类似于DOS中的 cls命令。清屏后，提示符移动到屏幕左上角。

例如：$ clear
当然$ reset也具有同样的效果!

____
碧轩
人之初，性本色!
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:19am 发表主题:

磁盘操作与管理[转帖]
在Linux系统中，如何有效地对存储空间加以使用和管理，是一项非常重要的技术。本章讲述如何查看系统中存储空间的使用情况、如何进行文件的转储、以及如何进行软盘的格式化。

磁盘空间管理

系统软件和应用软件，都要以文件的形式存储在计算机的磁盘空间中。因此，应该随时监视磁盘空间的使用情况。Linux系统提供了一组有关磁盘空间管理的命令。

df命令

功能：检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

语法：df ［选项］

说明：df命令可显示所有文件系统对i节点和磁盘块的使用情况。

该命令各个选项的含义如下：

-a 显示所有文件系统的磁盘使用情况，包括0块（block）的文件系统，如/proc文件系统。

-k 以k字节为单位显示。

-i 显示i节点信息，而不是磁盘块。

-t 显示各指定类型的文件系统的磁盘空间使用情况。

-x 列出不是某一指定类型文件系统的磁盘空间使用情况（与t选项相反）。

-T 显示文件系统类型。

例1：列出各文件系统的磁盘空间使用情况。

$ df

Filesystem 1 K－blocks Used Available Use% Mounted on

/dev/hda2 1361587 1246406 44823 97％ /

df命令的输出清单的第1列是代表文件系统对应的设备文件的路径名（一般是硬盘上的分区）；第2列给出分区包含的数据块（1024字节）的数目；第3，4列分别表示已用的和可用的数据块数目。用户也许会感到奇怪的是，第3，4列块数之和不等于第2列中的块数。这是因为缺省的每个分区都留了少量空间供系统管理员使用。即使遇到普通用户空间已满的情况，管理员仍能登录和留有解决问题所需的工作空间。清单中Use% 列表示普通用户空间使用的百分比，即使这一数字达到100％，分区仍然留有系统管理员使用的空间。最后，Mounted on列表示文件系统的安装点。

例2：列出各文件系统的i节点使用情况。

$ df -ia

Filesystem Inodes IUsed IFree Iused% Mounted on

/dev/ hda2 352256 75043 277213 21％ /

none 0 0 0 0% /proc

localhost:（pid221） 0 0 0 0% /net

例3：列出文件系统的类型。

$ df -T

Filesystem Type 1K－blocks Used Available use% Mounted on

/dev/hda2 ext2 1361587 1246405 44824 97% /

本例中的文件系统是ext2类型的。

du命令

du的英文原义为“disk usage”，含义为显示磁盘空间的使用情况。

功能：统计目录（或文件）所占磁盘空间的大小。

语法：du ［选项］ ［Names…］

说明：该命令逐级进入指定目录的每一个子目录并显示该目录占用文件系统数据块（1024字节）的情况。若没有给出Names，则对当前目录进行统计。

该命令的各个选项含义如下：

-s 对每个Names参数只给出占用的数据块总数。

-a 递归地显示指定目录中各文件及子孙目录中各文件占用的数据块数。若既不指定-s，也不指定-a，则只显示Names中的每一个目录及其中的各子目录所占的磁盘块数。

-b 以字节为单位列出磁盘空间使用情况（系统缺省以k字节为单位）。

-k 以1024字节为单位列出磁盘空间使用情况。

-c 最后再加上一个总计（系统缺省设置）。

-l 计算所有的文件大小，对硬链接文件，则计算多次。

-x 跳过在不同文件系统上的目录不予统计。

下面举例说明du命令的使用：

例1：查看/mnt目录占用磁盘空间的情况。 $ cd /mnt $ ls -lFR total 10 -rwxrwxrwx 2 root root 110 Jul 31 00:33 aa*

drwxr-xr-x 2 root root 1024 Jul 20 14:16 dev/

-rw-r--r-- 1 root root 6229 Aug 2 01:39 slack

drwxrwxrwx 2 root root 1024 Aug 2 02:09 var/

lrwxrwxrwx 1 root root 10 Aug 2 01:51 wei->/home/wei/

dev:

total 0

lrwxrwxrwx 1 root root 8 Jul 20 14:16 cdrom->/dev/hdb

var:

tatal 37

-r-xr-xr-x l root root 36064 Aug 2 02:09 rawrite. exe*

例2：列出各目录所占的磁盘空间，但不详细列出每个文件所占的空间。

$ du

l ./dev

38 ./var

48 .

输出清单中的第一列是以块为单位计的磁盘空间容量，第二列列出目录中使用这些空间的目录名称。

注意不带选项的du命令将从当前目录开始沿着目录结构向下工作直到列出所有目录的容量为止。这可能是一个很长的清单，有时只需要一个总数。这时可在du命令中加-s选项来取得总数：

$ du －s /mnt

/mnt

例3：列出所有文件和目录所占的空间（使用a选项），而且以字节为单位（使用b选项）来计算大小。

$ du -ab

8 ./dev/cdrom

1032 ./dev

36064 ./var/rawrite. exe

37088 ./var

6229 ./s1ack

110 ./aa

10 ./wei

45493 .

磁 盘 操 作

dd命令

功能：把指定的输入文件拷贝到指定的输出文件中，并且在拷贝过程中可以进行格式转换。可以用该命令实现DOS下的diskcopy命令的作用。先用dd命令把软盘上的数据写成硬盘的一个寄存文件，再把这个寄存文件写入第二张软盘上，完成diskcopy的功能。需要注意的是，应该将硬盘上的寄存文件用rm命令删除掉。系统默认使用标准输入文件和标准输出文件。

语法：dd ［选项］

if =输入文件（或设备名称）。

of =输出文件（或设备名称）。

ibs = bytes 一次读取bytes字节，即读入缓冲区的字节数。

skip = blocks 跳过读入缓冲区开头的ibs*blocks块。

obs = bytes 一次写入bytes字节，即写入缓冲区的字节数。

bs = bytes 同时设置读/写缓冲区的字节数（等于设置ibs和obs）。

cbs = byte 一次转换bytes字节。

count=blocks 只拷贝输入的blocks块。

conv = ASCII 把EBCDIC码转换为ASCIl码。

conv = ebcdic 把ASCIl码转换为EBCDIC码。

conv = ibm 把ASCIl码转换为alternate EBCDIC码。

conv = block 把变动位转换成固定字符。

conv = ublock 把固定位转换成变动位。

conv = ucase 把字母由小写转换为大写。

conv = lcase 把字母由大写转换为小写。

conv = notrunc 不截短输出文件。

conv = swab 交换每一对输入字节。

conv = noerror 出错时不停止处理。

conv = sync 把每个输入记录的大小都调到ibs的大小（用NUL填充）。

例1：要把一张软盘的内容拷贝到另一张软盘上，利用/tmp作为临时存储区。把源盘插入驱动器中，输入下述命令：

$ dd if =/dev/fd0 of = /tmp/tmpfile

拷贝完成后，将源盘从驱动器中取出，把目标盘插入，输入命令：

$ dd if = /tmp/tmpfile of =/dev/fd0

软盘拷贝完成后，应该将临时文件删除：

$ rm /tmp/tmpfile

例2：把net.i这个文件写入软盘中，并设定读/写缓冲区的数目。

（注意：软盘中的内容会被完全覆盖掉）

$ dd if = net.i of = /dev/fd0 bs = 16384

例3：将文件sfile拷贝到文件 dfile中。

$ dd if=sfile of=dfile

fdformat 命令

软盘是用户常用的存储介质之一。软盘在使用之前必须先作格式化操作，然后可以用tar、dd、cpio等命令存储数据，也可以在软盘上建立可安装的文件系统。

功能：低级格式化软盘

语法：format ［-n］ device

说明：该命令的作用是对软盘进行格式化。

-n 软盘格式化后不作检验。

device 指定要进行格式化的设备，通常是下述设备之一：

/dev/fd0d360

/dev/fd0h1200

/dev/fd0D360

/dev/fd0H360

/dev/fd0D720

/dev/fd0H720

/dev/fd0h360

/dev/fd0h720

/dev/fd0H1440

____

碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:20am 发表主题:

进程调度

kill命令

当需要中断一个前台进程的时候，通常是使用组合键；但是对于一个后台进程恐怕就不是一个组合键所能解决的了，这时就必须求助于kill命令。该命令可以终止后台进程。至于终止后台进程的原因很多，或许是该进程占用的CPU时间过多；或许是该进程已经挂死。总之这种情况是经常发生的。

kill命令是通过向进程发送指定的信号来结束进程的。如果没有指定发送信号，那么默认值为TERM信号。TERM信号将终止所有不能捕获该信号的进程。至于那些可以捕获该信号的进程可能就需要使用kill（9）信号了，该信号是不能被捕捉的。

kill命令的语法格式很简单，大致有以下两种方式：

kill ［-s 信号 | -p ］ ［ -a ］ 进程号 ...

kill -l ［信号］

-s 指定需要送出的信号。既可以是信号名也可以对应数字。

-p 指定kill命令只是显示进程的pid，并不真正送出结束信号。

-l 显示信号名称列表，这也可以在/usr/include/linux/signal.h文件中找到。

kill命令的使用

下面看看该命令的使用。

［例24］ 在执行一条find指令时由于时间过长，决定终止该进程。

首先应该使用ps命令来查看该进程对应的PID，键入ps，显示如下：

PID TTY TIME COMMAND

285 1 00:00:00 -bash

287 3 00:00:00 -bash

289 5 00:00:00 /sbin/mingetty tty5

290 6 00:00:00 /sbin/mingetty tty6

312 3 00:00:00 telnet bbs3

341 4 00:00:00 /sbin/mingetty tty4

345 1 00:00:00 find / -name foxy.jpg

348 1 00:00:00 ps

可以看到该进程对应的PID是345，现在使用kill命令来终止该进程。键入：

kill 345

再用ps命令查看，就可以看到，find进程已经被杀掉了。

［例25］ 杀掉进程11721

ps

PID TTY TIME COMMAND

11668 p1 00:00:00 -tcsh

11721 p1 00:00:00 cat

11737 p1 00:00:00 ps

kill 11721

［1］ Terminated cat

#

有时候可能会遇到这样的情况，某个进程已经挂死或闲置，使用kill命令却杀不掉。这时候就必须发送信号9，强行关闭此进程。当然这种“野蛮”的方法很可能会导致打开的文件出现错误或者数据丢失之类的错误。所以不到万不得已不要使用强制结束的办法。如果连信号9都不响应，那恐怕就只有重新启动计算机了。

控制多个进程命令

Linux可使用户同时运行多个进程，还允许用户或系统管理员能控制正在运行的进程。

nohup命令

理论上，我们一般退出Linux系统时，会把所有的程序全部结束掉，包括那些后台程序。但有时候，例如您正在编辑一个很长的程序，但是您下班或是有事需要先退出系统，这时您又不希望系统把您编辑那么久的程序结束掉，希望退出系统时，程序还能继续执行。这时，我们就可以使用nohup命令使进程在用户退出后仍继续执行。

一般这些进程我们都是让它在后台执行，结果则会写到用户自己的目录下的nohup.out这个文件里（也可以使用输出重定向，让它输出到一个特定的文件）。［例26］

$ nohup sort sales.dat &

这条命令告诉sort命令忽略用户已退出系统，它应该一直运行，直到进程完成。利用这种方法，可以启动一个要运行几天甚至几周的进程，而且在它运行时，用户不需要去登录。

nohup命令把一条命令的所有输出和错误信息送到nohup.out文件中。若将输出重定向，则只有错误信息放在nohup.out文件中。

renice命令

renice命令允许用户修改一个正在运行进程的优先权。

利用renice命令可以在命令执行时调整其优先权。其格式如下：

$ renice -number PID

其中，参数number与nice命令的number意义相同。

注：

（1） 用户只能对自己所有的进程使用renice命令。

（2） root用户可以在任何进程上使用renice命令。

（3） 只有root用户才能提高进程的优先权。

____
碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:22am 发表主题:

进程查看[转帖]
本节中要介绍的不只是进程查看方面的内容，由于Linux是个多用户系统，有时候也要了解其他用户现在在干什么，所以在本节中还将接触多用户方面的内容。同时Linux是一个多进程系统，经常需要对这些进程进行一些调配和管理；而要进行管理，首先就要知道现在的进程情况：究竟有哪些进程？进程情况如何？等等。所以需要进程查看方面的工作。

who命令

该命令主要用于查看当前在线上的用户情况。这个命令非常有用。如果用户想和其他用户建立即时通讯，比如使用talk命令，那么首先要确定的就是该用户确实在线上，不然talk进程就无法建立起来。又如，系统管理员希望监视每个登录的用户此时此刻的所作所为，也要使用who命令。

who命令的常用语法格式如下：

who ［imqsuwHT］ ［--count］ ［--idle］ ［--heading］ ［--help］ ［--message］ ［--mesg］ ［--version］ ［--writable］ ［file］ ［am i］

所有的选项都是可选的，也就是说可以单独使用who命令。不使用任何选项时，who命令将显示以下三项内容：

login name：登录用户名；

terminal line：使用终端设备；

login time：登录到系统的时间。

如果给出的是两个非选项参数，那么who命令将只显示运行who程序的用户名、登录终端和登录时间。通常这两个参数是“am i”，即该命令格式为：“who am i”。

下面对who命令的常用参数进行说明。

-m 和“who am i”的作用一样，显示运行该程序的用户名。

-q，--count 只显示用户的登录帐号和登录用户的数量，该选项优先级高于其他任何选项。

-s 忽略。主要是用于和其他版本的who命令兼容。

-i，-u，--idle 在登录时间后面显示该用户最后一次对系统进行操作至今的时间，也就是常说的“发呆”时间。其中“.”符号代表该用户在前1秒仍然处于活动状态；“old”则表示该用户空闲已经超过了24小时。

-H，--heading 显示一行列标题。常用的标题如表4-2所示。

表4-2　who命令输出常用标题

标 题

说 明

USER

用户登录帐号

LINE

用户登录使用终端

LOGIN-TIME

用户登录时间

IDLE

用户空闲时间，即未进行操作的时间

PID

用户登录shell的进程ID

FROM

用户网络地址

-w，-T--mesg，--message，--writable 和-s选项一样，在登录帐号后面显示一个字符来表示用户的信息状态：

+：允许写信息；

-：不允许写信息；

?：不能找到终端设备。

--help 在标准输出上显示帮助信息。

--version 在标准输出上显示版本信息。

下面介绍who命令的一些基本用法。

如果需要查看在系统上究竟有哪些用户，可以直接使用who命令。

［例17］ 查看登录到系统的用户情况

$ who

root tty1 Mar 17 13:49

foxy tty2 Mar 17 13:49

root tty3 Mar 17 13:49

bbs ttyp0 Mar 17 13:49 （river.net）

可以看到，现在系统一共有四个用户。第一列是登录用户的帐号；第二列是登录所使用的终端；第三列是登录时间；第四列是用户从什么地方登录的网络地址，这里是域名。

一般来说，这样就可以了解登录用户的大致情况了。但有时上面的显示不是那么直观，因为没有标题说明，不容易看懂，这时就需要使用-H选项了。

［例18］ 查看登录用户的详细情况，键入：

$ who -uH

显示如下：

USER LINE LOGIN-TIME IDLE FROM

root tty1 Mar 17 13:49 .

foxy tty2 Mar 17 13:49 00:01

root tty3 Mar 17 13:49 00:01

bbs ttyp0 Mar 17 13:49 00:01 （river.net）

这样一目了然。其中-u选项指定显示用户空闲时间，所以可以看到多了一项IDLE。第一个root用户的IDLE项是一个“.”，这就说明该用户在前1秒仍然是活动的，而其他用户后面都有一个时间，称为空闲时间。

最后来看看使用“who am i”格式命令的结果：

233.river.net!root tty1 Mar 17 13:49

可见只显示出了运行该who命令的用户情况，当然这时候不存在空闲时间。

who命令应用起来非常简单，可以比较准确地掌握用户的情况，所以使用非常广泛。

w命令

该命令也用于显示登录到系统的用户情况，但是与who不同的是，w命令功能更加强大，它不但可以显示

有谁登录到系统，还可以显示出这些用户当前正在进行的工作，并且统计数据相对who命令来说更加详细和科学，可以认为w命令就是who命令的一个增强版。

w命令的显示项目按以下顺序排列：当前时间，系统启动到现在的时间，登录用户的数目，系统在最近1秒、5秒和15秒的平均负载。然后是每个用户的各项数据，项目显示顺序如下：登录帐号、终端名称、远程主机名、登录时间、空闲时间、JCPU、PCPU、当前正在运行进程的命令行。

其中JCPU时间指的是和该终端（tty）连接的所有进程占用的时间。这个时间里并不包括过去的后台作业时间，但却包括当前正在运行的后台作业所占用的时间。而PCPU时间则是指当前进程（即在WHAT项中显示的进程）所占用的时间。下面介绍该命令的具体用法和参数。

语法格式如下：

w －［husfV］ ［user］

下面对参数进行说明：

-h 不显示标题。

-u 当列出当前进程和CPU时间时忽略用户名。这主要是用于执行su命令后的情况。

-s 使用短模式。不显示登录时间、JCPU和PCPU时间。

-f 切换显示FROM项，也就是远程主机名项。默认值是不显示远程主机名，当然系统管理员可以对源文件作一些修改使得显示该项成为默认值。

-V 显示版本信息。

User 只显示指定用户的相关情况。

［例19］ 显示当前登录到系统的用户的详细情况

$ w

2:50pm up 2 min， 4 users， load average:0.22，0.16，0.06

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root tty1 2:49pm 0:00s 0.56s 0.10s w

foxy tty2 2:49pm 1:09 0.42s 0.42s bash

root tty3 2:49pm 46.00s 0.67s 0.25s telnet bbs3

bbs ttyp0 river.net 2:49pm 45.00s 0.49s 0.49s bbs h river.net

ps命令

1. ps简介

前面介绍的两个命令都是用于查看当前系统用户的情况，下面就来看看进程的情况，这也是本章的主题。要对进程进行监测和控制，首先必须要了解当前进程的情况，也就是需要查看当前进程，而ps命令就是最基本同时也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等。总之大部分信息都是可以通过执行该命令得到的。

2. ps命令及其参数

ps命令最常用的还是用于监控后台进程的工作情况，因为后台进程是不和屏幕键盘这些标准输入/输出设备进行通信的，所以如果需要检测其情况，便可以使用ps命令了。

该命令语法格式如下：

ps ［选项］

下面对命令选项进行说明：

-e 显示所有进程。

-f 全格式。

-h 不显示标题。

-l 长格式。

-w 宽输出。

a 显示终端上的所有进程，包括其他用户的进程。

r 只显示正在运行的进程。

x 显示没有控制终端的进程。

O［+|-］ k1 ［，［+|-］ k2 ［，…］］ 根据SHORT KEYS、k1、k2中快捷键指定的多级排序顺序显示进程列表。对于ps的不同格式都存在着默认的顺序指定。这些默认顺序可以被用户的指定所覆盖。其中“+”字符是可选的，“-”字符是倒转指定键的方向。

pids 只列出指定进程的情况。各进程ID之间使用逗号分隔。该进程列表必须在命令行参数的最后一个选项后面紧接着给出，中间不能插入空格。比如：ps -f1,4,5。

以下介绍长命令行选项，这些选项都使用“--”开头：

--sort X［+|-］ key ［，［+|-］ key ［，…］］ 从SORT KEYS段中选一个多字母键。“+”字符是可选的，因为默认的方向就是按数字升序或者词典顺序。比如： ps -jax -sort=uid，-ppid，+pid。

--help 显示帮助信息。

--version 显示该命令的版本信息。

在前面的选项说明中提到了排序键，接下来对排序键作进一步说明。需要注意的是排序中使用的值是ps使用的内部值，并非仅用于某些输出格式的伪值。排序键列表见表4-3。

表4-3　排序键列表

短格式

长格式

说 明

c

cmd

可执行的简单名称

C

cmdline

完整命令行

f

flags

长模式标志

g

pgrp

进程的组ID

G

tpgid

控制tty进程组ID

j

cutime

累计用户时间

J

cstime

累计系统时间

k

utime

用户时间

K

stime

系统时间

m

min_flt

次要页错误的数量

M

maj_flt

主要页错误的数量

n

cmin_flt

累计次要页错误

N

cmaj_flt

累计主要页错误

o

session

对话ID

p

pid

进程ID

P

ppid

父进程ID

r

rss

驻留大小

R

resident

驻留页

s

size

内存大小（千字节）

S

share

共享页的数量

t

tty

tty次要设备号

T

start_time

进程启动的时间

U

uid

UID

u

user

用户名

v

vsize

总的虚拟内存数量（字节）

y

priority

内核调度优先级

3. 常用ps命令参数

前面两节介绍的参数可能让读者觉得有些可怕，实际上这是一个非常容易使用的命令，一般的用户只需掌握一些最常用的命令参数就可以了。

最常用的三个参数是u、a、x，下面将通过例子来说明其具体用法。

［例20］ 以root身份登录系统，查看当前进程状况

$ ps

PID TTY TIME COMMAND

5800 ttyp0 00:00:00 bash

5835 ttyp0 00:00:00 ps

可以看到，显示的项目共分为四项，依次为PID（进程ID）、TTY（终端名称）、TIME（进程执行时间）、COMMAND（该进程的命令行输入）。

可以使用u选项来查看进程所有者及其他一些详细信息，如下所示：

$ ps u

USER PID %CPU %MEM USZ RSS TTY STAT START TIME COMMAND

test 5800 0.0 0.4 1892 1040 ttyp0 S Nov27 0:00 -bash

test 5836 0.0 0.3 2528 856 ttyp0 R Nov27 0:00 ps u

在bash进程前面有条横线，意味着该进程便是用户的登录shell，所以对于一个登录用户来说带短横线的进程只有一个。还可以看到%CPU、%MEM两个选项，前者指该进程占用的CPU时间和总时间的百分比；后者指该进程占用的内存和总内存的百分比。

在这种情况下看到了所有控制终端的进程；但是对于其他那些没有控制终端的进程还是没有观察到，所以这时就需要使用x选项。使用x选项可以观察到所有的进程情况。

［例21］ 下面是使用x选项的例子：

$ ps x

PID TTY STAT TIME COMMAND

5800 ttyp0 S 0:00 -bash

5813 ttyp1 S 0:00 -bash

5921 ttyp0 S 0:00 man ps

5922 ttyp0 S 0:00 sh -c /usr/bin/gunzip -c /var/catman/cat1/ps.1.gz | /

5923 ttyp0 S 0:00 /usr/bin/gunzip -c /var/catman/cat1/ps.1.gz

5924 ttyp0 S 0:00 /usr/bin/less -is

5941 ttyp1 R 0:00 ps x

可以发现突然一下子就多出了那么多的进程。这些多出来的进程就是没有控制终端的进程。

前面看到的所有进程都是test用户自己的。其实还有许多其他用户在使用着系统，自然也就对应着其他的很多进程。如果想对这些进程有所了解，可以使用a选项来查看当前系统所有用户的所有进程。经常使用的是aux组合选项，这可以显示最详细的进程情况。

［例22］

```
$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 1136 64 ? S Nov25 0:02 init ［3］
root 2 0.0 0.0 0 0 ? SW Nov25 0:00 ［kflushd］
root 3 0.0 0.0 0 0 ? SW Nov25 0:03 ［kupdate］
root 4 0.0 0.0 0 0 ? SW Nov25 0:00 ［kpiod］
root 5 0.0 0.0 0 0 ? SW Nov25 0:00 ［kswapd］
root 163 0.0 0.1 1628 332 ? S Nov25 0:02 sshd
root 173 0.0 0.0 1324 200 ? S Nov25 0:00 syslogd
root 181 0.0 0.0 1420 0 ? SW Nov25 0:00 ［klogd］
daemon 191 0.0 0.1 1160 312 ? S Nov25 0:00 /usr/sbin/atd
root 201 0.0 0.1 1348 492 ? S Nov25 0:00 crond
root 212 0.0 0.0 1292 68 ? S Nov25 0:00 inetd
……
```

在显示的最前面是其他用户的进程情况，可以看到有root、daemon等用户以及他们所启动的进程。

在上面的例子中，介绍了ps命令最常见的一些选项和选项组合，用户可以根据自己的需要选用。

top命令

top命令和ps命令的基本作用是相同的，显示系统当前的进程和其他状况；但是top是一个动态显示过程，即可以通过用户按键来不断刷新当前状态。如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。

比较准确的说，top命令提供了实时的对系统处理器的状态监视。它将显示系统中CPU最“敏感”的任务列表。该命令可以按CPU使用。内存使用和执行时间对任务进行排序；而且该命令的很多特性都可以通过交互式命令或者在个人定制文件中进行设定。在后面的介绍中将把命令参数和交互命令分开讲述。

下面是该命令的语法格式：

top ［-］ ［d delay］ ［q］ ［c］ ［s］ ［S］ ［i］

d 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变之。

q 该选项将使top没有任何延迟的进行刷新。如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。

S 指定累计模式。

s 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。

i 使top不显示任何闲置或者僵死进程。

c 显示整个命令行而不只是显示命令名

top命令显示的项目很多，默认值是每5秒更新一次，当然这是可以设置的。显示的各项目为：

uptime 该项显示的是系统启动时间、已经运行的时间和三个平均负载值（最近1秒，5秒，15秒的负载值）。

processes 自最近一次刷新以来的运行进程总数。当然这些进程被分为正在运行的，休眠的，停止的等很多种类。进程和状态显示可以通过交互命令t来实现。

CPU states 显示用户模式，系统模式，优先级进程（只有优先级为负的列入考虑）和闲置等各种情况所占用CPU时间的百分比。优先级进程所消耗的时间也被列入到用户和系统的时间中，所以总的百分比将大于100%。

Mem 内存使用情况统计，其中包括总的可用内存，空闲内存，已用内存，共享内存和缓存所占内存的情况。

Swap 交换空间统计，其中包括总的交换空间，可用交换空间，已用交换空间。

PID 每个进程的ID。

PPID 每个进程的父进程ID。

UID 每个进程所有者的UID 。

USER 每个进程所有者的用户名。

PRI 每个进程的优先级别。

NI 该进程的优先级值。

SIZE 该进程的代码大小加上数据大小再加上堆栈空间大小的总数。单位是KB。

TSIZE 该进程的代码大小。对于内核进程这是一个很奇怪的值。

DSIZE 数据和堆栈的大小。

TRS 文本驻留大小。

D 被标记为“不干净”的页项目。

LIB 使用的库页的大小。对于ELF进程没有作用。

RSS 该进程占用的物理内存的总数量，单位是KB。

SHARE 该进程使用共享内存的数量。

STAT 该进程的状态。其中S代表休眠状态；D代表不可中断的休眠状态；R代表运行状态；Z代表僵死状态；T代表停止或跟踪状态。

TIME 该进程自启动以来所占用的总CPU时间。如果进入的是累计模式，那么该时间还包括这个进程子进程所占用的时间。且标题会变成CTIME。

%CPU 该进程自最近一次刷新以来所占用的CPU时间和总时间的百分比。

%MEM 该进程占用的物理内存占总内存的百分比。

COMMAND 该进程的命令名称，如果一行显示不下，则会进行截取。内存中的进程会有一个完整的命令行。

下面介绍在top命令执行过程中可以使用的一些交互命令。从使用角度来看，熟练的掌握这些命令比掌握选项还重要一些。这些命令都是单字母的，如果在命令行选项中使用了s选项，则可能其中一些命令会被屏蔽掉。

〈空格〉 立即刷新显示。

Ctrl+L 擦除并且重写屏幕。

h或者? 显示帮助画面，给出一些简短的命令总结说明。

k 终止一个进程。系统将提示用户输入需要终止的进程PID，以及需要发送给该进程什么样的信号。一般的终止进程可以使用15信号；如果不能正常结束那就使用信号9强制结束该进程。默认值是信号15。在安全模式中此命令被屏蔽。

i 忽略闲置和僵死进程。这是一个开关式命令。

q 退出程序。

r 重新安排一个进程的优先级别。系统提示用户输入需要改变的进程PID以及需要设置的进程优先级值。输入一个正值将使优先级降低，反之则可以使该进程拥有更高的优先权。默认值是10。

S 切换到累计模式。

s 改变两次刷新之间的延迟时间。系统将提示用户输入新的时间，单位为s。如果有小数，就换算成ms。输入0值则系统将不断刷新，默认值是5 s。需要注意的是如果设置太小的时间，很可能会引起不断刷新，从而根本来不及看清显示的情况，而且系统负载也会大大增加。

f或者F 从当前显示中添加或者删除项目。

o或者O 改变显示项目的顺序。

l 切换显示平均负载和启动时间信息。

m 切换显示内存信息。

t 切换显示进程和CPU状态信息。

c 切换显示命令名称和完整命令行。

M 根据驻留内存大小进行排序。

P 根据CPU使用百分比大小进行排序。

T 根据时间/累计时间进行排序。

W 将当前设置写入~/.toprc文件中。这是写top配置文件的推荐方法。

从上面的介绍中可以看到，top命令是一个功能十分强大的监控系统的工具，尤其对于系统管理员而言更是如此。一般的用户可能会觉得ps命令其实就够用了，但是top命令的强劲功能确实提供了不少方便。下面来看看实际使用的情况。

［例23］ 键入top命令查看系统状况

$ top

1：55pm up 7 min， 4 user， load average:0.07，0.09，0.06

29 processes:28 sleeping， 1 running， 0 zombie， 0 stopped

CPU states: 4.5% user， 3.6% system， 0.0% nice， 91.9% idle

Mem: 38916K av， 18564K used， 20352K free， 11660K shrd， 1220K buff

Swap: 33228K av， 0K used， 33228K free， 11820K cached

PID USER PRI NI SIZE RSS SHARE STAT LIB %CPU %MEM TIME COMMAND

363 root 14 0 708 708 552 R 0 8.1 1.8 0:00 top

1 root 0 0 404 404 344 S 0 0.0 1.0 0:03 init

2 root 0 0 0 0 0 SW 0 0.0 0.0 0:00 kflushd

3 root -12 -12 0 0 0 SW< 0 0.0 0.0 0:00 kswapd

4 root 0 0 0 0 0 SW 0 0.0 0.0 0:00 md_thread

5 root 0 0 0 0 0 SW 0 0.0 0.0 0:00 md_thread

312 root 1 0 636 636 488 S 0 0.0 1.6 0:00 telnet

285 root 6 0 1140 1140 804 S 0 0.0 2.9 0.00 bash

286 root 0 0 1048 1048 792 S 0 0.0 2.6 0.00 bash

25 root 0 0 364 364 312 S 0 0.0 0.9 0.00 kerneld

153 root 0 0 456 456 372 S 0 0.0 1.1 0.00 syslogd

160 root 0 0 552 552 344 S 0 0.0 1.4 0.00 klogd

169 daemon 0 0 416 416 340 S 0 0.0 1.0 0.00 atd

178 root 2 0 496 496 412 S 0 0.0 1.2 0.00 crond

187 bin 0 0 352 352 284 S 0 0.0 0.9 0.00 portmap

232 root 0 0 500 500 412 S 0 0.0 1.2 0.00 rpc.mountd

206 root 0 0 412 412 344 S 0 0.0 1.0 0.00 inetd

215 root 0 0 436 436 360 S 0 0.0 1.1 0.00 icmplog

第一行的项目依次为当前时间、系统启动时间、当前系统登录用户数目、平均负载。第二行为进程情况，依次为进程总数、休眠进程数、运行进程数、僵死进程数、终止进程数。第三行为CPU状态，依次为用户占用、系统占用、优先进程占用、闲置进程占用。第四行为内存状态，依次为平均可用内存、已用内存、空闲内存、共享内存、缓存使用内存。第五行为交换状态，依次为平均可用交换容量、已用容量、闲置容量、高速缓存容量。然后下面就是和ps相仿的各进程情况列表了。

总的来说，top命令的功能强于ps，但需要长久占用前台，所以用户应该根据自己的情况来使用这个命令。

碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:24am 发表主题:

进程管理及作业控制
进程管理及作业控制(一)

Linux是一个多任务的操作系统，系统上同时运行着多个进程，正在执行的一个或多个相关进程称为一个作业。使用作业控制，用户可以同时运行多个作业，并在需要时在作业之间进行切换。本章详细介绍进程管理及作业控制的命令，包括启动进程、查看进程、调度作业的命令。

进程及作业的概念

Linux是一个多用户多任务的操作系统。多用户是指多个用户可以在同一时间使用计算机系统；多任务是指Linux可以同时执行几个任务，它可以在还未执行完一个任务时又执行另一项任务。

操作系统管理多个用户的请求和多个任务。大多数系统都只有一个CPU和一个主存，但一个系统可能有多个二级存储磁盘和多个输入/输出设备。操作系统管理这些资源并在多个用户间共享资源，当您提出一个请求时，给您造成一种假象，好象系统只被您独自占用。而实际上操作系统监控着一个等待执行的任务队列，这些任务包括用户作业、操作系统任务、邮件和打印作业等。操作系统根据每个任务的优先级为每个任务分配合适的时间片，每个时间片大约都有零点几秒，虽然看起来很短，但实际上已经足够计算机完成成千上万的指令集。每个任务都会被系统运行一段时间，然后挂起，系统转而处理其他任务；过一段时间以后再回来处理这个任务，直到某个任务完成，从任务队列中去除。

Linux系统上所有运行的东西都可以称之为一个进程。每个用户任务、每个系统管理守护进程，都可以称之为进程。Linux用分时管理方法使所有的任务共同分享系统资源。我们讨论进程的时候，不会去关心这些进程究竟是如何分配的，或者是内核如何管理分配时间片的，我们所关心的是如何去控制这些进程，让它们能够很好地为用户服务。

进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。进程与程序是有区别的，进程不是程序，虽然它由程序产生。程序只是一个静态的指令集合，不占系统的运行资源；而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。而且一个程序可以启动多个进程。

Linux操作系统包括三种不同类型的进程，每种进程都有自己的特点和属性。

*

交互进程——由一个shell启动的进程。交互进程既可以在前台运行，也可以在后台运行。
*

批处理进程——这种进程和终端没有联系，是一个进程序列。
*

监控进程（也称守护进程）——Linux系统启动时启动的进程，并在后台运行。

上述三种进程各有各的作用，使用场合也有所不同。

进程和作业的概念也有区别。一个正在执行的进程称为一个作业，而且作业可以包含一个或多个进程，尤其是当使用了管道和重定向命令。例如“nroff -man ps.1|grep kill|more”这个作业就同时启动了三个进程。

作业控制指的是控制正在运行的进程的行为。比如，用户可以挂起一个进程，等一会儿再继续执行该进程。shell将记录所有启动的进程情况，在每个进程过程中，用户可以任意地挂起进程或重新启动进程。作业控制是许多shell（包括bash和tcsh）的一个特性，使用户能在多个独立作业间进行切换。

一般而言，进程与作业控制相关联时，才被称为作业。

在大多数情况下，用户在同一时间只运行一个作业，即它们最后向shell键入的命令。但是使用作业控制，用户可以同时运行多个作业，并在需要时在这些作业间进行切换。这会有什么用途呢？例如，当用户编辑一个文本文件，并需要中止编辑做其他事情时，利用作业控制，用户可以让编辑器暂时挂起，返回shell提示符开始做其他的事情。其他事情做完以后，用户可以重新启动挂起的编辑器，返回到刚才中止的地方，就象用户从来没有离开编辑器一样。这只是一个例子，作业控制还有许多其他实际的用途。

启动进程

键入需要运行的程序的程序名，执行一个程序，其实也就是启动了一个进程。在Linux系统中每个进程都具有一个进程号，用于系统识别和调度进程。启动一个进程有两个主要途径：手工启动和调度启动，后者是事先进行设置，根据用户要求自行启动。

手工启动

由用户输入命令，直接启动一个进程便是手工启动进程。但手工启动进程又可以分为很多种，根据启动的进程类型不同、性质不同，实际结果也不一样，下面分别介绍。

1. 前台启动

这或许是手工启动一个进程的最常用的方式。一般地，用户键入一个命令“ls －l”，这就已经启动了一个进程，而且是一个前台的进程。这时候系统其实已经处于一个多进程状态。或许有些用户会疑惑：我只启动了一个进程而已。但实际上有许多运行在后台的、系统启动时就已经自动启动的进程正在悄悄运行着。还有的用户在键入“ls －l”命令以后赶紧使用“ps －x”查看，却没有看到ls进程，也觉得很奇怪。其实这是因为ls这个进程结束太快，使用ps查看时该进程已经执行结束了。如果启动一个比较耗时的进程:

find / -name fox.jpg

然后再把该进程挂起，使用ps查看，就会看到一个find进程在里面。

2. 后台启动

直接从后台手工启动一个进程用得比较少一些，除非是该进程甚为耗时，且用户也不急着需要结果的时候。假设用户要启动一个需要长时间运行的格式化文本文件的进程。为了不使整个shell在格式化过程中都处于“瘫痪”状态，从后台启动这个进程是明智的选择。

［例1］

$ troff －me notes ＞ note_form &

［1］ 4513

$

由上例可见，从后台启动进程其实就是在命令结尾加上一个&号。键入命令以后，出现一个数字，这个数字就是该进程的编号，也称为PID，然后就出现了提示符。用户可以继续其他工作。

上面介绍了前、后台启动的两种情况。实际上这两种启动方式有个共同的特点，就是新进程都是由当前shell这个进程产生的。也就是说，是shell创建了新进程，于是就称这种关系为进程间的父子关系。这里shell是父进程，而新进程是子进程。一个父进程可以有多个子进程，一般地，子进程结束后才能继续父进程；当然如果是从后台启动，那就不用等待子进程结束了。

一种比较特殊的情况是在使用管道符的时候。例如：

nroff -man ps.1|grep kill|more

这时候实际上是同时启动了三个进程。请注意是同时启动的，所有放在管道两边的进程都将被同时启动，它们都是当前shell的子程序，互相之间可以称为兄弟进程。

以上介绍的是手工启动进程的一些内容，作为一名系统管理员，很多时候都需要把事情安排好以后让其自动运行。因为管理员不是机器，也有离开的时候，所以有些必须要做的工作而恰好管理员不能亲自操作，这时候就需要使用调度启动进程了。

调度启动

有时候需要对系统进行一些比较费时而且占用资源的维护工作，这些工作适合在深夜进行，这时候用户就可以事先进行调度安排，指定任务运行的时间或者场合，到时候系统会自动完成这一切工作。

要使用自动启动进程的功能，就需要掌握以下几个启动命令。

at命令

用户使用at命令在指定时刻执行指定的命令序列。也就是说，该命令至少需要指定一个命令、一个执行时间才可以正常运行。at命令可以只指定时间，也可以时间和日期一起指定。需要注意的是，指定时间有个系统判别问题。比如说：用户现在指定了一个执行时间：凌晨3:20，而发出at命令的时间是头天晚上的20:00，那么究竟是在哪一天执行该命令呢？如果用户在3:20以前仍然在工作，那么该命令将在这个时候完成；如果用户3:20以前就退出了工作状态，那么该命令将在第二天凌晨才得到执行。下面是at命令的语法格式：

at ［-V］ ［-q 队列］ ［-f 文件名］ ［-mldbv］ 时间

at -c 作业 ［作业...］

at允许使用一套相当复杂的指定时间的方法，实际上是将POSIX.2标准扩展了。它可以接受在当天的hh:mm（小时:分钟）式的时间指定。如果该时间已经过去，那么就放在第二天执行。当然也可以使用midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午4点）等比较模糊的词语来指定时间。用户还可以采用12小时计时制，即在时间后面加上AM（上午）或者PM（下午）来说明是上午还是下午。

也可以指定命令执行的具体日期，指定格式为month day（月 日）或者mm/dd/yy（月/日/年）或者dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。

上面介绍的都是绝对计时法，其实还可以使用相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为：now + count time-units ，now就是当前时间，time-units是时间单位，这里可以是minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count是时间的数量，究竟是几天，还是几小时，等等。

还有一种计时方法就是直接使用today（今天）、tomorrow（明天）来指定完成命令的时间。下面通过一些例子来说明具体用法。

［例2］ 指定在今天下午5:30执行某命令。假设现在时间是中午12:30，1999年2月24日，其命令格式如下：

at 5:30pm

at 17:30

at 17:30 today

at now + 5 hours

at now + 300 minutes

at 17:30 24.2.99

at 17:30 2/24/99

at 17:30 Feb 24

以上这些命令表达的意义是完全一样的，所以在安排时间的时候完全可以根据个人喜好和具体情况自由选择。一般采用绝对时间的24小时计时法可以避免由于用户自己的疏忽造成计时错误的情况发生，例如上例可以写成：

at 17:30 2/24/99

这样非常清楚，而且别人也看得懂。

对于at命令来说，需要定时执行的命令是从标准输入或者使用-f选项指定的文件中读取并执行的。如果at命令是从一个使用su命令切换到用户shell中执行的，那么当前用户被认为是执行用户，所有的错误和输出结果都会送给这个用户。但是如果有邮件送出的话，收到邮件的将是原来的用户，也就是登录时shell的所有者。

［例3］

$ at -f work 4pm + 3 days

在三天后下午4点执行文件work中的作业。

$ at -f work 10am Jul 31

在7月31日上午10点执行文件work中的作业。

在任何情况下，超级用户都可以使用这个命令。对于其他用户来说，是否可以使用就取决于两个文件：/etc/at.allow和/etc/at.deny。如果/etc/at.allow文件存在的话，那么只有在其中列出的用户才可以使用at命令；如果该文件不存在，那么将检查/etc/at.deny文件是否存在，在这个文件中列出的用户均不能使用该命令。如果两个文件都不存在，那么只有超级用户可以使用该命令；空的/etc/at.deny文件意味着所有的用户都可以使用该命令，这也是默认状态。

下面对命令中的参数进行说明。

-V 将标准版本号打印到标准错误中。

-q queue 使用指定的队列。队列名称是由单个字母组成，合法的队列名可以由a-z或者A-Z。a队列是at命令的默认队列。

-m 作业结束后发送邮件给执行at命令的用户。

-f file 使用该选项将使命令从指定的file读取，而不是从标准输入读取。

-l atq命令的一个别名。该命令用于查看安排的作业序列，它将列出用户排在队列中的作业，如果是超级用户，则列出队列中的所有工作。

命令的语法格式如下:

atq ［-V］ ［-q 队列］ ［-v］

-d atrm 命令的一个别名。该命令用于删除指定要执行的命令序列，语法格式如下：

atrm ［-V］ 作业 ［作业...］

-c 将命令行上所列的作业送到标准输出。

［例4］ 找出系统中所有以txt为后缀名的文件，并且进行打印。打印结束后给用户foxy发出邮件通知取件。指定时间为十二月二十五日凌晨两点。

首先键入：

$ at 2:00 12/25/99

然后系统出现at>提示符，等待用户输入进一步的信息，也就是需要执行的命令序列：

at> find / -name “*.txt”|lpr

at> echo “foxy：All texts have been printed.You can take them over.Good day!River” |mail -s ”job done” foxy

输入完每一行指令然后回车，所有指令序列输入完毕后，使用<Ctrl+d>组合键结束at命令的输入。这时候屏幕将出现如下信息：

warning:command will be executed using /bin/sh.

job 1 at 1999-12-25 02:00

提醒用户将使用哪个shell来执行该命令序列。

实际上如果命令序列较长或者经常被执行的时候，一般都采用将该序列写到一个文件中，然后将文件作为at命令的输入来处理。这样不容易出错。

［例5］ 上面的例子可以修改如下：

将命令序列写入到文件/tmp/printjob，语句为：

$ at -f /tmp/printjob 2:00 12/25/99

这样一来，at命令将使用文件中的命令序列，屏幕显示如下：

Warning:command will be executed using /bin/sh.

job 2 at 1999-12-25 02:00

当然也可以采用以下命令：

$ at< /tmp/printjob 2:00 12/25/99

来完成同样的任务。也就是使用输入重定向的办法将文件定向为命令输入。

batch命令

batch 用低优先级运行作业，该命令几乎和at命令的功能完全相同，唯一的区别在于，at命令是在指定时间，很精确的时刻执行指定命令；而batch却是在系统负载较低，资源比较空闲的时候执行命令。该命令适合于执行占用资源较多的命令。

batch命令的语法格式也和at命令十分相似，即

batch ［-V］ ［-q 队列］ ［-f 文件名］ ［-mv］ ［时间］

具体的参数解释请参考at命令。一般地说，不用为batch命令指定时间参数，因为batch本身的特点就是由系统决定执行任务的时间，如果用户再指定一个时间，就失去了本来的意义。

［例6］ 使用例4，键入：

$ batch

at> find / -name *.txt|lpr

at> echo “foxy：All texts have been printed.You can take them over.Good day!River” |mail -s ”job done” foxy

现在这个命令就会在合适的时间进行了，进行完后会发回一个信息。

仍然使用<Ctrl+d>组合键来结束命令输入。而且batch和at命令都将自动转入后台，所以启动的时候也不需要加上&符号。

cron命令

前面介绍的两条命令都会在一定时间内完成一定任务，但是要注意它们都只能执行一次。也就是说，当指定了运行命令后，系统在指定时间完成任务，一切就结束了。但是在很多时候需要不断重复一些命令，比如：某公司每周一自动向员工报告头一周公司的活动情况，这时候就需要使用cron命令来完成任务了。

实际上，cron命令是不应该手工启动的。cron命令在系统启动时就由一个shell脚本自动启动，进入后台（所以不需要使用&符号）。一般的用户没有运行该命令的权限，虽然超级用户可以手工启动cron，不过还是建议将其放到shell脚本中由系统自行启动。

首先cron命令会搜索/var/spool/cron目录，寻找以/etc/passwd文件中的用户名命名的crontab文件，被找到的这种文件将载入内存。例如一个用户名为foxy的用户，它所对应的crontab文件就应该是/var/spool/cron/foxy。也就是说，以该用户命名的crontab文件存放在/var/spool/cron目录下面。cron命令还将搜索/etc/crontab文件，这个文件是用不同的格式写成的。

cron启动以后，它将首先检查是否有用户设置了crontab文件，如果没有就转入“休眠”状态，释放系统资源。所以该后台进程占用资源极少。它每分钟“醒”过来一次，查看当前是否有需要运行的命令。命令执行结束后，任何输出都将作为邮件发送给crontab的所有者，或者是/etc/crontab文件中MAILTO环境变量中指定的用户。

上面简单介绍了一些cron的工作原理，但是cron命令的执行不需要用户干涉；需要用户修改的是crontab中要执行的命令序列，所以下面介绍crontab命令。

crontab命令

crontab命令用于安装、删除或者列出用于驱动cron后台进程的表格。也就是说，用户把需要执行的命令序列放到crontab文件中以获得执行。每个用户都可以有自己的crontab文件。下面就来看看如何创建一个crontab文件。

在/var/spool/cron下的crontab文件不可以直接创建或者直接修改。crontab文件是通过crontab命令得到的。现在假设有个用户名为foxy，需要创建自己的一个crontab文件。首先可以使用任何文本编辑器建立一个新文件，然后向其中写入需要运行的命令和要定期执行的时间。

然后存盘退出。假设该文件为/tmp/test.cron。再后就是使用crontab命令来安装这个文件，使之成为该用户的crontab文件。键入：

crontab test.cron

这样一个crontab 文件就建立好了。可以转到/var/spool/cron目录下面查看，发现多了一个foxy文件。这个文件就是所需的crontab 文件。用more命令查看该文件的内容可以发现文件头有三行信息：

#DO NOT EDIT THIS FILE -edit the master and reinstall.

#（test.cron installed on Mon Feb 22 14:20:20 1999）

#（cron version --$Id:crontab.c，v 2.13 1994/01/17 03:20:37 vivie Exp $）

大概意思是：

#切勿编辑此文件——如果需要改变请编辑源文件然后重新安装。

#test.cron文件安装时间：14:20:20 02/22/1999

如果需要改变其中的命令内容时，还是需要重新编辑原来的文件，然后再使用crontab命令安装。

可以使用crontab命令的用户是有限制的。如果/etc/cron.allow文件存在，那么只有其中列出的用户才能使用该命令；如果该文件不存在但cron.deny文件存在，那么只有未列在该文件中的用户才能使用crontab命令；如果两个文件都不存在，那就取决于一些参数的设置，可能是只允许超级用户使用该命令，也可能是所有用户都可以使用该命令。

crontab命令的语法格式如下：

crontab ［-u user］ file

crontab ［-u user］{-l|-r|-e}

第一种格式用于安装一个新的crontab 文件，安装来源就是file所指的文件，如果使用“-”符号作为文件名，那就意味着使用标准输入作为安装来源。

-u 如果使用该选项，也就是指定了是哪个具体用户的crontab 文件将被修改。如果不指定该选项，crontab 将默认是操作者本人的crontab ，也就是执行该crontab 命令的用户的crontab 文件将被修改。但是请注意，如果使用了su命令再使用crontab 命令很可能就会出现混乱的情况。所以如果是使用了su命令，最好使用-u选项来指定究竟是哪个用户的crontab文件。

-l 在标准输出上显示当前的crontab。

-r 删除当前的crontab文件。

-e 使用VISUAL或者EDITOR环境变量所指的编辑器编辑当前的crontab文件。当结束编辑离开时，编辑后的文件将自动安装。

［例7］

crontab -l #列出用户目前的crontab。

10 6 * * * date

0 */2 * * * date

0 23-7/2，8 * * * date

#

在crontab文件中如何输入需要执行的命令和时间。该文件中每行都包括六个域，其中前五个域是指定命令被执行的时间，最后一个域是要被执行的命令。每个域之间使用空格或者制表符分隔。格式如下：

minute hour day-of-month month-of-year day-of-week commands

第一项是分钟，第二项是小时，第三项是一个月的第几天，第四项是一年的第几个月，第五项是一周的星期几，第六项是要执行的命令。这些项都不能为空，必须填入。如果用户不需要指定其中的几项，那么可以使用*代替。因为*是统配符，可以代替任何字符，所以就可以认为是任何时间，也就是该项被忽略了。在表4-1中给出了每项的合法范围。

表4-1　指定时间的合法范围

时间	合法值
minute	00-59
hour	00-23，其中00点就是晚上12点
day-of-month	01-31
month-of-year	01-12
day-of-week	0-6，其中周日是0

这样用户就可以往crontab 文件中写入无限多的行以完成无限多的命令。命令域中可以写入所有可以在命令行写入的命令和符号，其他所有时间域都支持列举，也就是域中可以写入很多的时间值，只要满足这些时间值中的任何一个都执行命令，每两个时间值中间使用逗号分隔。

［例8］

5，15，25，35，45，55 16，17，18 * * * command

这就是表示任意天任意月，其实就是每天的下午4点、5点、6点的5 min、15 min、25 min、35 min、45 min、55 min时执行命令。

［例9］ 在每周一，三，五的下午3：00系统进入维护状态，重新启动系统。那么在crontab 文件中就应该写入如下字段：

00 15 * * 1，3，5 shutdown -r +5

然后将该文件存盘为foxy.cron，再键入crontab foxy.cron安装该文件。

［例10］ 每小时的10分，40分执行用户目录下的innd/bbslin这个指令：

10，40 * * * * innd/bbslink

［例11］ 每小时的1分执行用户目录下的bin/account这个指令：

1 * * * * bin/account

［例12］ 每天早晨三点二十分执行用户目录下如下所示的两个指令（每个指令以;分隔）：

20 3 * * * （/bin/rm -f expire.ls logins.bad;bin/expire>expire.1st）

［例13］ 每年的一月和四月，4号到9号的3点12分和3点55分执行/bin/rm -f expire.1st这个指令，并把结果添加在mm.txt这个文件之后（mm.txt文件位于用户自己的目录位置）。

12,55 3 4-9 1,4 * /bin/rm -f expire.1st>>mm.txt

［例14］ 我们来看一个超级用户的crontab文件：

#Run the ‘atrun’ program every minutes

#This runs anything that’s due to run from ‘at’.See man ‘at’ or ‘atrun’.

0,5,10,15,20,25,30,35,40,45,50,55 * * * * /usr/lib/atrun

40 7 * * * updatedb

8,10,22,30,39,46,54,58 * * * * /bin/sync

进程的挂起及恢复命令bg、fg

作业控制允许将进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按<ctrl+z>，即可挂起当前的前台作业。

［例15］

$ cat > text.file

<ctrl+z>

［1］ + stopped cat > text.file

$ jobs

［1］+ stopped cat > text.file

在键盘上按<ctrl+z>后，将挂起当前执行的命令cat。使用jobs命令可以显示shell的作业清单，包括具体的作业、作业号以及作业当前所处的状态。

恢复进程执行时，有两种选择：用fg命令将挂起的作业放回到前台执行；用bg命令将挂起的作业放到后台执行。

［例16］ 用户正在使用Emacs，突然需要查看系统进程情况。就首先使用<Ctrl+z>组合键将Emacs进程挂起，然后使用bg命令将其在后台启动，这样就得到了前台的操作控制权，接着键入“ps －x”查看进程情况。查看完毕后，使用fg命令将Emacs带回前台运行即可。其命令格式为：

<Ctrl+z>

$ bg emacs

$ ps －x

$ fg emacs

默认情况下，fg和bg命令对最近停止的作业进行操作。如果希望恢复其他作业的运行，可以在命令中指定要恢复作业的作业号来恢复该作业。例如：

$ fg 1

cat > text.file

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:25am 发表主题:

在Linux环境下运行DOS命令

Linux系统提供了一组称为mtools的可移植工具，可以让用户轻松地从标准的DOS软盘上读、写文件和目录。它们对DOS和Linux环境之间交换文件非常有用。它们是不具备共同的文件系统格式的系统之间交换文件的有力手段。对于一个MS-DOS的软盘，只要把软盘放在软驱中，就可以利用mtools提供的命令来访问软盘上的文件。

mtools的主要命令如下：

mcd 目录名 改变MSDOS目录；

mcopy 源文件 目标文件 在MSDOS和Unix之间复制文件；

mdel 文件名 删除MSDOS文件；

mdir 目录名 显示MSDOS目录；

mformat 驱动器号 在低级格式化的软盘上创建MSDOS文件系统；

rnlabel 驱动器号 产生MSDOS卷标；

mmd 目录名 建立MSDOS目录；

mrd 目录名 删除MSDOS目录；

mren 源文件 目标文件 重新命名已存在的MSDOS文件；

mtype 文件名 显示MSDOS文件的内容。

这些命令和对应的不加 m的 MSDOS命令非常相似。

例1：在Linux环境下看DOS盘最上层的目录的内容：

$ mdir a:

Volume in drive A has no label

Volume Serial Number is 15F6-3362

Directory of A:

SS6 CPP 331 09-24-99 7:41 ss6.cpp

CH9 ＜DIR＞ 11-20-99 16:22 ch9

XXQ 0 11-20-99 16:24 xxq

95CZXTA DOC 36,864 06-15-98 22:51 95czxta.doc

95CZXTB DOC 39,936 06-16-98 7:18 95czxtb.doc

HTCA DOC 27,136 01-08-99 0:13 htca.doc

HTCB DOC 27,136 01-08-99 0:12 htcb.doc

6 file（s） 131,403 bytes

1 dir（s） 1,295,872 bytes free

例2：将DOS盘上的文件xxq复制到当前目录下，并用ls命令进行验证。

$ mcopy a:htca.doc

$ ls －l htca.doc

-rw-r— -r- - 1 xxq xxq 27136 Jan 1 01:80 htca.doc

思 考 题

1. 如何找到用户主目录的绝对路径名？在自己的系统上，用户主目录的绝对路径名是什么？

2. 将当前工作目录从/home/xxq转到/home/wang需要使用什么命令？

3. 如何显示当前目录？

4. 如何在当前目录下建立子目录text？

5. 如何删除子目录text？

6. 如何查看当前目录下的内容？

7. 如何将文件text的权限设定为：text属主可读、可写、可执行，同组用户可读、可执行，其他用

户可执行？

8. 如何将当前目录包括所有子目录全部做备份文件，备份文件名为first.tar？

9. 如何将目录/home下每一个文件压缩成.gz文件？

10. 如何把上例中每个压缩的文件解压，并列出详细的信息？

11. 如何将当前目录下的text文件拷贝到DOS软盘上，并用什么命令可以进行验证？

12. 如何将DOS盘上的example文件复制到当前目录下，并用什么命令可以进行验证？

13. 假设你以普通用户身份访问系统，显示文件系统中所有文件名为core的文件清单。

14. TurboLiunx系统提供了哪些有关文件显示的命令，它们从功能上来说有何区别？

15. 如何将目录/home/xxq下的文件text复制到目录/home/wang下，并将文件名重新命名为example，最后将目录/home/xxq下的文件text删除。

____
碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:26am 发表主题:

备份与压缩命令
　　用户经常需要备份计算机系统中的数据，为了节省存储空间，常常将备份文件进行压缩。下面分别介绍备份与压缩的命令。

　　tar命令

tar可以为文件和目录创建档案。利用tar，用户可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件。tar最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案，如软盘。利用tar命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。Linux上的tar是GNU版本的。

语法：tar ［主选项+辅选项］ 文件或者目录

使用该命令时，主选项是必须要有的，它告诉tar要做什么事情，辅选项是辅助使用的，可以选用。

主选项：

c 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。

r 把要存档的文件追加到档案文件的未尾。例如用户已经作好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将忘记的目录或文件追加到备份文件中。

t 列出档案文件的内容，查看已经备份了哪些文件。

u 更新文件。就是说，用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。

x 从档案文件中释放文件。

辅助选项：

b 该选项是为磁带机设定的。其后跟一数字，用来说明区块的大小，系统预设值为20（20*512 bytes）。

f 使用档案文件或设备，这个选项通常是必选的。

k 保存已经存在的文件。例如我们把某个文件还原，在还原的过程中，遇到相同的文件，不会进行覆盖。

m 在还原文件时，把所有文件的修改时间设定为现在。

M 创建多卷的档案文件，以便在几个磁盘中存放。

v 详细报告tar处理的文件信息。如无此选项，tar不报告文件信息。

w 每一步都要求确认。

z 用gzip来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，但还原时也一定要使用该选项进行解压缩。

例1：把/home目录下包括它的子目录全部做备份文件，备份文件名为usr.tar。

$ tar cvf usr.tar /home

例2：把/home目录下包括它的子目录全部做备份文件，并进行压缩，备份文件名为usr.tar.gz 。

$ tar czvf usr.tar.gz /home

例3：把usr.tar.gz这个备份文件还原并解压缩。

$ tar xzvf usr.tar.gz

例4：查看usr.tar备份文件的内容，并以分屏方式显示在显示器上。

$ tar tvf usr.tar | more

要将文件备份到一个特定的设备，只需把设备名作为备份文件名。

例5：用户在/dev/fd0设备的软盘中创建一个备份文件，并将/home 目录中所有的文件都拷贝到备份文件中。

$ tar cf /dev/fd0 /home

要恢复设备磁盘中的文件，可使用xf选项：

$ tar xf /dev/fd0

如果用户备份的文件大小超过设备可用的存贮空间，如软盘，您可以创建一个多卷的tar备份文件。M选项指示tar命令提示您使用一个新的存贮设备，当使用M选项向一个软驱进行存档时，tar命令在一张软盘已满的时候会提醒您再放入一张新的软盘。这样您就可以把tar档案存入几张磁盘中。

$ tar cMf /dev/fd0 /home

要恢复几张盘中的档案，只要将第一张放入软驱，然后输入有x和M选项的tar命令。在必要时您会被提醒放入另外一张软盘。

$ tar xMf /dev/fd0

gzip命令

减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip是在Linux系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。

语法：gzip ［选项］ 压缩（解压缩）的文件名

各选项的含义：

-c 将输出写到标准输出上，并保留原有文件。

-d 将压缩文件解压。

-l 对每个压缩文件，显示下列字段：

压缩文件的大小

未压缩文件的大小

压缩比

未压缩文件的名字

-r 递归式地查找指定目录并压缩其中的所有文件或者是解压缩。

-t 测试，检查压缩文件是否完整。

-v 对每一个压缩和解压的文件，显示文件名和压缩比。

-num 用指定的数字num调整压缩的速度，-1或--fast表示最快压缩方法（低压缩比），-9或--best表示最慢压缩方法（高压缩比）。系统缺省值为6。

假设一个目录/home下有文件mm.txt、sort.txt、xx.com。

例1：把/home目录下的每个文件压缩成.gz文件。

$ cd /home

$ gzip *

$ ls

m.txt.gz sort.txt.gz xx.com.gz

例2：把例1中每个压缩的文件解压，并列出详细的信息。

$ gzip -dv *

mm.txt.gz 43.1%-----replaced with mm.txt

sort.txt.gz 43.1%-----replaced with sort.txt

xx.com.gz 43.1%-----replaced with xx.com

$ ls

mm.txt sort.txt xx.com

例3：详细显示例1中每个压缩的文件的信息，并不解压。

$ gzip -l *

compressed uncompr. ratio uncompressed_name

277 445 43.1% mm.txt

278 445 43.1% sort.txt

277 445 43.1% xx.com

$ ls

mm.txt.gz sort.txt.gz xx.com.gz

例4：压缩一个tar备份文件，如usr.tar，此时压缩文件的扩展名为.tar.gz

$ gzip usr.tar

$ ls

usr.tar.gz

unzip命令

用MS Windows下的压缩软件winzip压缩的文件如何在Linux系统下展开呢？可以用unzip命令，该命令用于解扩展名为.zip的压缩文件。

语法：unzip ［选项］ 压缩文件名.zip

各选项的含义分别为：

-x 文件列表 解压缩文件，但不包括指定的file文件。

-v 查看压缩文件目录，但不解压。

-t 测试文件有无损坏，但不解压。

-d 目录 把压缩文件解到指定目录下。

-z 只显示压缩文件的注解。

-n 不覆盖已经存在的文件。

-o 覆盖已存在的文件且不要求用户确认。

-j 不重建文档的目录结构，把所有文件解压到同一目录下。

例1：将压缩文件text.zip在当前目录下解压缩。

$ unzip text.zip

例2：将压缩文件text.zip在指定目录/tmp下解压缩，如果已有相同的文件存在，要求unzip命令不覆盖原先的文件。

$ unzip -n text.zip -d /tmp

例3：查看压缩文件目录，但不解压。

$ unzip -v text.zip

zgrep命令

这个命令的功能是在压缩文件中寻找匹配的正则表达式，用法和grep命令一样，只不过操作的对象是压缩文件。如果用户想看看在某个压缩文件中有没有某一句话，便可用zgrep命令。

____
碧轩
人之初，性本色！
http://www.ourlinux.net/
http://yum.ourlinux.net/
http://apt.ourlinux.net/
http://download.ourlinux.net/
游戏交易：http://www.0579tt.com/

返回页首

bixuan
公社特别嘉宾

注册: Nov 23, 2002
文章: 1994
位置: 浙江金华
发表于: 2003-04-12, 9:27am 发表主题:

文件链接命令

ln命令

该命令在文件之间创建链接。这种操作实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称。对于这个新的文件名，我们可以为之指定不同的访问权限，以控制对信息的共享和安全性的问题。

如果链接指向目录，用户就可以利用该链接直接进入被链接的目录而不用打一大堆的路径名。而且，即使我们删除这个链接，也不会破坏原来的目录。

语法：ln ［选项］ 目标 ［链接名］

ln ［选项］ 目标 目录

链接有两种，一种被称为硬链接（Hard Link），另一种被称为符号链接（Symbolic Link）。建立硬链接时，链接文件和被链接文件必须位于同一个文件系统中，并且不能建立指向目录的硬链接。而对符号链接，则不存在这个问题。默认情况下，ln产生硬链接。

在硬链接的情况下，参数中的“目标”被链接至［链接名］。如果［链接名］是一个目录名，系统将在该目录之下建立一个或多个与“目标”同名的链接文件，链接文件和被链接文件的内容完全相同。如果［链接名］为一个文件，用户将被告知该文件已存在且不进行链接。如果指定了多个“目标”参数，那么最后一个参数必须为目录。

如果给ln命令加上- s选项，则建立符号链接。如果［链接名］已经存在但不是目录，将不做链接。［链接名］可以是任何一个文件名（可包含路径），也可以是一个目录，并且允许它与“目标”不在同一个文件系统中。如果［链接名］是一个已经存在的目录，系统将在该目录下建立一个或多个与“目标”同名的文件，此新建的文件实际上是指向原“目标”的符号链接文件。

例：$ ln – s lunch /home/xu

用户为当前目录下的文件lunch创建了一个符号链接/home/xu。

上一页 目录 下一页

上一页 目录 下一页

轻松安装、卸载Linux软件

以下三篇文单大部分内容是雷同的，懒得整理，所以全放上来了！！
第一篇：

Linux下软件的安装与卸载

在Windows下安装软件时，只需运行软件的安装程序（setup、install等）或者用zip等解压缩软件解开即可安装，运行反安装程序（uninstall、unware、“卸载”等）就能将软件清除干净，完全图形化的操作界面，简单到只要用鼠标一直点击“下一步”就可以了。而Linux好象就不一样了，很多的初学者都抱怨在Linux下安装和卸载软件非常地困难，没有像使用Windows时那么直观。其实在Linux下安装和卸载软件也非常简单，同样也有安装向导或解压安装的方式，不相同的只不过是除了二进制形式的软件分发外，还有许许多多以源代码形式分发的软件包，下面就来详细地讲一讲这些软件的安装与卸载：

一、二进制分发软件包的安装与卸载

Linux软件的二进制分发是指事先已经编译好二进制形式的软件包的发布形式，其优点是安装使用容易，缺点则是缺乏灵活性，如果该软件包是为特定的硬件/操作系统平台编译的，那它就不能在另外的平台或环境下正确执行。

1、*.rpm形式的二进制软件包

安装：rpm -ivh *.rpm

卸载：rpm -e packgename

说明：RPM（RedHat Packge Manager）是RedHat公司出的软件包管理器，使用它可以很容易地对rpm形式的软件包进行安装、升级、卸载、验证、查询等操作，安装简单，而卸载时也可以将软件安装在多处目录中的文件删除干净，因此推荐初学者尽可能使用rpm形式的软件包。rpm的参数中-i是安装，-v是校验，-h是用散列符显示安装进度，*.rpm是软件包的文件名（这里的*.rpm特指*.src.rpm以外的以rpm为后缀的文件）；参数-e是删除软件包，packgename是软件包名，与软件包的文件名有所区别，它往往是文件名中位于版本号前面的字符串，例如apache-3.1.12-i386.rpm和apache-devel-3.1.12-i386.rpm是软件包文件名，它们的软件包名称分别是apache和apache-devel。更多的rpm参数请自行参看手册页：man rpm。

如果你不喜欢在字符界面下安装或卸载这些软件包，完全可以在X-Window下使用图形界面的软件包管理程序，如glint、xrpm这样的图形接口，或者是KDE的kpackge等，这样对软件包的安装、升级、卸载、验证和查询就可以通过点击鼠标来轻松完成。

2、*.tar.gz/*.tgz、*.bz2形式的二进制软件包

安装：tar zxvf *.tar.gz 或 tar yxvf *.bz2

卸载：手动删除

说明：*.tar.gz/*.bz2形式的二进制软件包是用tar工具来打包、用gzip/bzip2压缩的，安装时直接解包即可。对于解压后只有单一目录的软件，卸载时用命令“rm -rf 软件目录名”；如果解压后文件分散在多处目录中，则必须一一手动删除（稍麻烦），想知道解压时向系统中安装了哪些文件，可以用命令“tar ztvf *.tar.gz”/“tar ytvf *.bz2”获取清单。tar的参数z是调用gzip解压，x是解包，v是校验，f是显示结果，y是调用bzip2解压，t是列出包的文件清单。更多的参数请参看手册页：man tar。

如果你更喜欢图形界面的操作，可以在X-Window下使用KDE的ArK压缩档案管理工具。

3、提供安装程序的软件包

这类软件包已经提供了安装脚本或二进制的安装向导程序（setup、install、install.sh等），只需运行它就可以完成软件的安装；而卸载时也相应地提供了反安装的脚本或程序。例如SUN公司的StarOffice办公软件套件就使用名为setup的安装程序，而且在软件安装后提供反安装的功能，目前这种类型的软件包还比较少，因其安装与卸载的方式与Windows软件一样，所以就无需多讲了。

二、源代码分发软件包的安装与卸载

Linux软件的源代码分发是指提供了该软件所有程序源代码的发布形式，需要用户自已编译成可执行的二进制代码并进行安装，其优点是配置灵活，可以随意去掉或保留某些功能/模块，适应多种硬件/操作系统平台及编译环境，缺点是难度较大，一般不适合初学者使用。

1、*.src.rpm形式的源代码软件包

安装：rpm -rebuild *.src.rpm

cd /usr/src/dist/RPMS
rpm -ivh *.rpm
卸载：rpm -e packgename
说明：rpm --rebuild *.src.rpm命令将源代码编译并在/usr/src/dist/RPMS下生成二进制的rpm包，然后再安装该二进制包即可。packgename如前所述。
2、*.tar.gz/*.tgz、*.bz2形式的源代码软件包
安装：tar zxvf *.tar.gz 或 tar yxvf *.bz2 先解压
然后进入解压后的目录：
./configure 配置
make 编译
make install 安装
卸载：make uninstall 或 手动删除
说明：建议解压后先阅读说明文件，可以了解安装有哪些需求，有必要时还需改动编译配置。有些软件包的源代码在编译安装后可以用make install命令来进行卸载，如果不提供此功能，则软件的卸载必须手动删除。由于软件可能将文件分散地安装在系统的多个目录中，往往很难把它删除干净，那你应该在编译前进行配置，指定软件将要安装到目标路径：./configure --prefix=目录名，这样可以使用“rm -rf 软件目录名”命令来进行干净彻底的卸载。与其它安装方式相比，需要用户自己编译安装是最难的，它适合于使用Linux已有一定经验的人，一般不推荐初学者使用。

关于Linux下软件的安装与卸载lanche已经讲了这么多，但可能还会有人问怎么知道一个tar.gz/bz2包是二进制文件包呢还是源代码包？如果你用过压缩工具就会明白，压缩包未必就是软件，它也可能是备份的许多图片，也可能是打包在一起的普通资料，要分辨它到底是什么最好的办法就是查看包里的文件清单，使用命令tar ztvf *.tar.gz / tar ytvf *.bz2或者在X-Window下使用图形化的ArK压缩档案管理工具都可以，源代码包里的文件往往会含有种种源代码文件，头文件*.h、c代码源文件*.c、C++代码源文件*.cc/*.cpp等；而二进制包里的文件则会有可执行文件（与软件同名的往往是主执行文件），标志是其所在路径含有名为bin的目录（仅有少数例外）。原来这么简单呀，还不快点自己试试！

第二篇：

如何安装Linux的软件

在windows下安装软件大家都觉得很容易，只要双击setup或是install的图标，然后跟着向导一步一步的按提示做就可以了，但是在linux下安装软件就不像windows下那样容易了，有时你找到的是没有编译过的软件源码，那就更加的麻烦了，这里就介绍一下如何安装linux的软件啦！
linux下的软件一般都是经过压缩的，主要的格式有这几种：rpm、tar、tar.gz、tgz等。所以首先拿到软件后第一件事就是解压缩。
一、在xwindow下以rpm格式的软件安装比较容易，只要在把鼠标移到文件上单击右键，在弹出的菜单里会有专门的三项（只有在右键单击rpm文件才会出现）show info，upgrade和install，这三项的意思大家都很清楚了，我就不多说了。rpm格式说了，接着就是tar，tar.gz，tgz等，在xwindow下双击这些格式的文件就会自动解压缩的，是解压缩而不是像rpm那样有install选项的，install文件会在你解压缩后才出现，不过这里我要先说一下，在你双击install前建议先好好看一下readme，因为你所要安装这些软件可能是没有编译的，所以看一下会比较好。
二、说完了xwindow下的安装和解压缩，接着说在文本模式下如何解压缩和安装，还有编译源码了。首先说rpm的，以root身份登陆后（用其他的身份登陆也可以但有些软件就不能被安装）键入rpm －i你所要解压缩的文件＋扩展名（当然是rpm）也不是很难吧，这里我要说的是“－i”意思是install，如果你想要卸载或是要执行其他的指令就打rpm －－help，看电脑给你的提示吧，为了照顾那些E文不太好的人我

这里就写几个常用的参数：－e：卸载相关的应用程序，－U（注意是大写）：升级软件包， －pql：列出rpm软件包的相关信息，－qf：查找指定文件属于哪个软件包。至于那些其他格式的软件则可以用gunzip，gzip，tar和unzip等指令来解压缩，然后再运行install。通过解包后会得到一些文件，然后进入刚才解压缩后文件目录用“ls －F －color”指令看一下所得到的文件，一般有两种情况：第一种是文件为已经编译好的程序（无须再编译了）用上面的察看文件命令会以带“＊”标记的亮绿色显示；第二种则是需要由你自己编译的源代码。得到的是源代码的话，就需要我们自己编译来得到可运行的程序。编译源代码听到编译是不是就吓到你了呢，其实说穿了也就加上几句话而已，对初学者是有些困难想用好的操作系统就需要学习嘛，等你学会了在MM前露一手的时候想想吧……，在编译之前先要认真的阅读一下readme文档，老鸟们就不需要了吧，等你阅读完了以后就执行
./configure
make
make install（只有Root身份才能执行此命令），编译完成之后便会在当前目录或src子目录下得到软件的可执行程序。
介绍完了，大家是不是觉的RPM的软件安装比较容易一点呢，但我个人意见初学者还是多选择RPM格式以外的软件，因为通过编译软件的源码可以为你以后编译内核打下一定的基础。（上海 张国荣）

第三篇：

轻松安装、卸载Linux软件

Linux软件的安装和卸载一直是困扰许多新用户的难题。在Windows中，我们可以使用软件自带的安装卸载程序或在控制面板中的“添加/删除程序”来实现。与其相类似，在Linux下有一个功能强大的软件安装卸载工具，名为RPM。它可以用来建立、安装、查询、更新、卸载软件。该工具是在命令行下使用的。在Shell的提示符后输入rpm，就可获得该命令的帮助信息。

软件的安装
Linux下软件的安装主要有两种不同的形式。第一种安装文件名为xxx.tar.gz；另一种安装文件名为xxx.i386.rpm。以第一种方式发行的软件多为以源码形式发送的；第二种方式则是直接以二进制形式发送的。

对于第一种，安装方法如下：

1.首先，将安装文件拷贝至你的目录中。例如，如果你是以root身份登录上的，就将软件拷贝至/root中。

#cp xxx.tar.gz /root

2.由于该文件是被压缩并打包的,应对其解压缩。命令为：

#tar xvzf filename.tar.gz

3.执行该命令后，安装文件按路径，解压缩在当前目录下。用ls命令可以看到解压缩后的文件。通常在解压缩后产生的文件中，有“Install”的文件。该文件为纯文本文件，详细讲述了该软件包的安装方法。

4.执行解压缩后产生的一个名为configure的可执行脚本程序。它是用于检查系统是否有编译时所需的

库，以及库的版本是否满足编译的需要等安装所需要的系统信息。为随后的编译工作做准备。命令为：#./configure。

5.检查通过后，将生成用于编译的MakeFile文件。此时，可以开始进行编译了。编译的过程视软件的规模和计算机性能的不同，所耗费的时间也不同。命令为：#make。

6.成功编译后，键入如下的命令开始安装：

#make install

7.安装完毕，应清除编译过程中产生的临时文件和配置过程中产生的文件。键入如下命令：

#make clean
#make distclean

至此，软件的安装结束。

对于第二种，其安装方法要简单得多。

同第一种方式一样，将安装文件拷贝至你的目录中。然后使用rpm来安装该文件。命令如下：

#rpm -i filename.i386.rpm

rpm将自动将安装文件解包，并将软件安装到缺省的目录下。并将软件的安装信息注册到rpm的数据库中。参数i的作用是使rpm进入安装模式。

软件的卸载
1.软件的卸载主要是使用rpm来进行的。卸载软件首先要知道软件包在系统中注册的名称。键入命令：

#rpm -q -a

即可查询到当前系统中安装的所有的软件包。

2.确定了要卸载的软件的名称，就可以开始实际卸载该软件了。键入如下命令即可卸载软件：

#rpm -e [package name]

参数e的作用是使rpm进入卸载模式。对名为[package name]的软件包进行卸载。由于系统中各个软件包之间相互有依赖关系。如果因存在依赖关系而不能卸载，rpm将给予提示并停止卸载。你可以使用如下的命令来忽略依赖关系，直接开始卸载：

#rpm -e [package name] -nodeps

忽略依赖关系的卸载可能会导致系统中其它的一些软件无法使用。

补充：

如果是以.bin结尾的二进制软件，可以用以下方法安装（以so-6_0-beta-bin-linux-zh-Tw1.bin为例）：

在so-6_0-beta-bin-linux-zh-Tw1.bin所在文件夹下运行模拟终端
输入：
./so-6_0-beta-bin-linux-zh-Tw1.bin
并按回车！

当然也可以输入：
./so 再用[Tab]键补全

（./表示当前目录，如果终端不在该软件所在目录下打开，则在软件名前尚需输入相应的路径。）

如果在图形界面，也可直接单击（或双击，视具体的鼠标设置而定）进行安装！
<hr>

上一页 目录 下一页

上一页 目录 下一页

设置和修改X-Window的显示模式

简短说明:

本文档的主要目的在于介绍手动配置 Linux 下 XF86 的配置方法, 主要是关于如何通过修改 XF86Config 文件调整显示器在图形模式下的显示分辨率和颜色深度, 以及如何通过建立符号连接修改 X Server. 另外提到一些跟图形界面有关的网友常问问题.

一. X Window 工作方式略谈

(如果你之关心如何对 X Widow 进行设置, 可以跳过这一节. 只是理解这一节所讲的知识对于理解下面的各设置步骤有一定帮助)

X Window 的工作方式跟 Microsoft Windows 有着本质的不同. MS Windows 的图形用户界面 (GUI) 是跟系统紧密相联的. 而 X Window 则不是, 它实际上是在系统核心 (kernel) 的上面运行的一个应用程序.

X Window 的运行分为四层. 最底层的是 X Server(服务器), 提供图形界面的驱动, 为 X Window 提供服务. 上面的一层是用于网上通信的网络协议 -- X 网络协议. 这部分使远程运行 X Window 成为可能. 只需要在服务器上运行一个 X Server, 而客户机 (Client) 上运行更上一层的程序, 则可以实现 X Widow 的远程运行. 再往上的一层是称作 Xlib 的低层函数接口, 介于网络和和基础系统于较高层的程序之间. 应用程序的实现是通过调用这一层的函数实现的. 最顶层

就是管理窗口的窗口管理器了，也就是一般所说的 WM (Window Manager). 在这
一层的软件是用户都比较常接触的，比如 fvwm, AfterStep, Enlightment 以及
WindowMaker 等等.

从上面的介绍来看，X Window 的运行是一种客户机/服务器(Client/Server)
的模式，服务器用于显示客运的应用程序，又被称为显示服务器(Display Server)
显示服务器位于硬件和客户机之间，它跟踪所有来自输入设备比如键盘，鼠标的
输入动作，经过处理后将其送回客户机. 这样，使用户甚至可以在 MS Windows
的机器上运行 X Client，截取并传送用户的输入，只是将 X Window 的屏幕输出
显示在用户的屏幕上. 客户机的输入和输出系统跟 X 服务器之间的通信都是遵守
X 协议的.

由于本文档的主要目的在于介绍如何设置，因此对 X Window 不再作进一步的
介绍. 更详细的资料可以参考有关的 HOWTO, man 页等等.

二. 关于 X Window 的设置

1. X Server

在 Linux 下，这些服务器的程序是存放在 /usr/X11R6/bin/ 下面，并且都由
XF86_ 打头，一般来说，会提供单显 Mono，标准 VGA 和 SVGA 的驱动，分
别命名为：XF86_Mono, XF86_VGA16 和 XF86_SVGA，另外可能还会有 S3 和其他一
些带有图形加速功能的显卡的驱动. 下面是作者本人机器上运行的 RedHat Linux
6.1 中所带有的 X Server 驱动：

XF86_3DLabs XF86_Mono XF86_I128 XF86_VGA16
XF86_P9000 XF86_8514 XF86_Mach32 XF86_S3
XF86_W32 XF86_AGX XF86_Mach64 XF86_S3V
XF86_FBDev XF86_Mach8 XF86_SVGA

如果你的 /usr/X11R6/bin 下面没有上面某个文件或者多了某个文件，不必在
意，这主要决定于你的机器的硬件. 其实一般只需要跟自己硬件相配套的驱动即可.
比如作者正在使用的机器上使用的是 3D Lab Permedia II 的芯片的显示卡，因此
只有 XF86_3DLabs 是有作用的.

如此之多种类的显示适配器驱动程序，系统当然需要一个固定和统一的方法去
访问它，一般说来，系统是使用一个名为 X 的符号连接去访问具体的 X Server 的.
这个 X 也同样是存在于 /usr/X11R6/bin/ 目录中，可以使用如下命令打印他的具
体指向：

```
ls -l /usr/X11R6/bin/X
```

得到的结果如：

```
lrwxrwxrwx 1 root root 8 Dec 14 03:40
/usr/X11R6/bin/X -> XF86_3DLab
```

前面的属性 "lrwxrwxrwx" 中第一个字母 l 表示它是一个连接, 在最后由箭头
指出它所指向的具体目标. (这里的换行是为了阅读的方便)

明白了这个, 在以后需要更换 X Server 的时候就可以通过改变这个联系来实现
了, 具体命令如下:

ln -sf /usr/X11R6/bin/X??????? /usr/X11R6/bin/X

其中问号代表的字母根据具体情况决定. 有些时候, 由于这个连接不正确, 导致
无法运行 X Window, 这时可以使它指向 XF86_VGA16, 使用 16 色的 VGA 模式进入
X Window, 注意如果需要运行 XF86Setup 程序 (是个图形界面的 X Window 配置工
具程序), 那么这个 XF86_VGA16 驱动一定要存在, 这是由于该驱动所提供的显示模式
是一般显示卡和显示器都支持的, 故该配置程序使用此 X Server 为驱动. 当用户的
显卡是比较特殊的型号时, 需要装特定的驱动程序, 也需要作此工作.

驱动程序的获得有多种途径, 一般的 Linux 发行版本都提供了丰富的驱动程序,
可以满足大部分要求, 比较特殊的驱动程序, 可以在网上获得. 当然现在 Linux 对硬
件的支持还不是很完善, 但这已经超出了本文档的访问, 不在作更多的表述.

2. 显示模式

显示模式包括分辨率, 扫描频率, 色深等等. 这也是网友在显示方面提问最多的
一个问题. 对于现在的显示卡和显示器, 一般都可以使用 1024 x 768 x 16 bit 色的
显示模式. 当然保险起见, 在使用下面所叙述的方法设置您所使用的显示模式的时候,
请确定您的显示卡和显示器能够达到该模式的要求, 否则可能会有损坏硬件的危险.

在各发行版本中, 都会提供一些设置 X Window 的工具软件, 如 Xconfigurator,
XF86Setup, XF86Config 等等, 这些程序无一例外的是提供了一个修改 X Window 的
配置文件 XF86Config 的简便方法. 当然使用起来比较方便, 但是对于一些需要灵活
调整的场合就不是非常适合, 需要手动调整配置文件中的参数达到最好效果. 您不必
一定这样做, 这里只是介绍这样一种方法, 使您可以在有疑问的时候方便地解决.

一般这个文件是存在于 /usr/X11R6/lib/X11/ 目录下, 在 /etc 或者 /etc/X11
目录中有它的一个符号连接. 它是一个文本文件, 可以使用 vi 对它进行修改. 下面
对该文件中各部分的作用做个详细的介绍. (如有遗漏, 欢迎提出)

在这个文件中有这样几个概念: 段 (Section) 和子段 (Subsection). 每个段或
子段分别给出了不同的参数, 格式分别为:

```
Section
......
Endsection

Section
Subsection
......
Endsubsection
```

Endsection

一个完整的配置文件中包含六个段，分别是 File（文件），SeverFlag（服务器特殊标志），Keyboard（键盘），Pointer（指示设备，一般是鼠标），Monitor（监视器），Device（显示适配器，即显卡），Screen（屏幕设置信息）.

File 段：这一段指示出一些运行 X Window 所需要的文件的位置，比如颜色的别名文件 rgb.txt（注意这里是不需要写出该文件的扩展名），比如：

RgbPath "/usr/X11R6/lib/X11/rgb"

如果您看过 /usr/X11R6/lib/X11/rgb.txt 的内容，你会很容易的知道它是干什么用的. 有了它，你可以使用 red1 代表 RGB 颜色值 255 0 0 等等.

接下来是字体文件的路径，这里的设置将影响到中文字体的输出. 这里对路径的设置可以是多个，例如下面的这样：

FontPath "unix/:-1"
FontPath "/usr/share/zhfont/X11:unscaled"
FontPath "/usr/share/zhfont/TTF"
FontPath "/usr/X11R6/lib/X11/fonts/local"
FontPath "/usr/X11R6/lib/X11/fonts/chinese"
FontPath "/usr/X11R6/lib/X11/fonts/TTF"
FontPath "/usr/X11R6/lib/X11/fonts/Type1"
FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
FontPath "/usr/X11R6/lib/X11/fonts/misc"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
FontPath "/usr/X11R6/lib/X11/fonts/100dpi"

在使用一个字体目录之前，需要使用 mkfontdir 命令把一条路径中的字体作个列表，在该目录中生成 font.dir 作为对字体的索引，如：

mkfontdir /usr/X11Ry/lib/X11/fonts/100dpi

具体该命令的用法，请使用 man 命令查看其 man page（手册页），或者会在本文档的后续版本中提供详细的说明.

ServerFlag 段：这个段指明了一些具体的参数，如下：

NoTrapSignals 这个标志将使调试信号对 X Server 不起作用. 调试信号会使 X Server "转储核心"，把内存中有关的 X Server 的信息存储到硬盘上，即所谓的 "Core dump". 这样的目的是便于对 X Server 进行调试.

DontZap 这个标志将使 X Server 收到 "Ctrl + Alt + Back Space" 键组合发出的信号时 *不* 停止工作. 当然一般是不建议使用的.

DontZoom 这个标志使 X Window 不能在各种图形模式之间进行切换. X
Window 的缺省状态是在用户设定了多组分辨率, 刷新频率和颜色的组
合的情况下, 运行 X Window 的过程中可以通过 Ctrl + '+' 和 Ctrl + '-' 两组键的组合在多个图形
模式之间进行动态切换.

Keyboard 段: 这个段指明了 X Window 环境中所使用的键盘映射以及其他一些参
数. 如下是作者的键盘配置:

```
Section "Keyboard"
Protocol "Standard"
AutoRepeat 500 5 # 指定延迟 (delay) 和 速率 (rate)
# 这里的设定分别为 500 和 0
LeftAlt Meta
RightAlt Meta # 使用左右 Alt 键作 Meta 键.
# 关于 Meta 键, 希望以后能够补充完整
ScrollLock Compose
RightCtl Control
XkbKeycodes "xfree86"
XkbTypes "default"
XkbCompat "default"
XkbSymbols "us(pc101)"
XkbGeometry "pc"
XkbRules "xfree86"
XkbModel "pc101"
XkbLayout "us"
EndSection
```

这段设置出了 XkbLayout 会影响键盘各按键的表现以外, 其他都只是一些功
能上的设置, 没有太大影响. 在控制台模式下, 可以通过 Alt + F? 在各虚
拟终端间进行切换, 在 X Widow 环境下, 必须使用 Ctrl + Alt + F? 才能
够达到这一效果.

对于本段以及下面的 Pointer 段, 由于本人经验和所知的限制不能完整地讲
述, 会在不久以后作进一步补充. 并在此向各位表示歉意.

Pointer 段: 如前所述, 这个段是设置鼠标等设备的. 如下举例是作者的鼠标设
置:

```
Section "Pointer"
Protocol "PS/2"
Device "/dev/mouse"
# 下面这两个参数是一些采用磁技术的鼠标所需要的, 一般注释掉或者删除
# BaudRate 9600
# SampleRate 150
# 下面的两个参数是为方便使用双键鼠标的用户使用三键鼠标而设置的
# 当选择了虚拟三键的时候, 可以通过同时点击鼠标左右键完成中键的功能
Emulate3Buttons
```

Emulate3Timeout 50
下面这个参数也是为磁金属鼠标所设定的
ChordMiddle
EndSection

Mornitor 段: 这个段的设置需要参考显示器的用户手册, 以便获得相关的数据, 防止损坏显示器. 这个段是我改动最多的, 现举例描述如下:

Section "Monitor"
Identifier "Samsung SyncMaster 500s/500Ms"
监视器标示符, 在下面 Screen 段用作标示
这个参数的设置对显示器的模式没有直接影响, 只要保证跟后面
Screen 段的引用保持一致
VendorName "Unknown"
生产厂家的名称
ModelName "Unknown"
制造和型号的标示, 并不重要
在使用 Xconfigurator 的时候, 如果不提供上述值, 则填写
"Unknown"

下面这两个参数在设置显示模式上来书哦是比较重要的.
HorizSync 是水平扫描频率, kHz 为单位.
HorizSync 50-60
VertRefresh 是垂直扫描频率, Hz 为单位
VertRefresh 80-100
上面这两个参数可以是列举的形式, 也可以是范围的形式.
比如: 列举: 75 85 100
范围: 如上所述

接下来是模式的描述了, 如果您使用的是 TurboLinux 或者是
RedHat Linux, 下面的模式列表将会非常的完整. 但 Slackware
没有提供完整的列表, 这些数据可以在显示器的说明书上见到.
为节省空间, 这里仅列举一条
800x600 @ 85 Hz, 55.84 kHz hsync
Modeline "800x600" 60.75 800 864 928 1088 600 616 621 657 -HSync -VSync
或者写成以下形式, 为 Slackware 采用
Mode "800x600"
DotClock 45
HTimings 1024 1048 1208 1264
VTimings 768 776 784 817
Flags "Interlace"
EndMode
Endsection

这里来讲讲上面的 ModeLine 的涵义. 就象第二种形式所描述的那样, 这些参数是使用 "800x600x85Hz" 模式所需要的. 要设置显示为这种模式, 首先要保证水平扫描频率和垂直扫描频率要配合好, 即 VertRefresh 的范围或列

举值中要包括 85 Hz, 而 HorizSync 的值需要包括 55.84 kHz. 这些都可以
通过手工调整. 然后需要看看打点频率. 有些显示器有这个参数提供, 而另
外一些可能检测不到, 这时需要查看手册. 打点频率的值可以不提供, 这点
在下面会有讲述.

注意, 这里的 "800x600" 就是这种模式的标示. 等到下面谈到模式的设置顺
序就会清楚 X Window 是如何设置的了.

Device 段: 这里主要记录了显卡的数据, 如下可以看到, 很简单, 只有显存的大
小和显卡的时钟频率. 这个时钟频率是用来产生视频信号, 而显示器的扫描
频率也是根据这个信号产生的. 所以不要为了追求高分辨率和高刷新频率把
它调得太高, 否则可能会损伤显示器. 一般在显示器的说明书上会有这个值,
如果没有, 建议不要超过 50.

```
Section "Device"
Identifier "Generic VGA"
VendorName "Unknown"
BoardName "Unknown"
Chipset "generic"
# VideoRam 256
# Clocks 25.2 28.3
EndSection
```

注意: 上面所说的两个段都可以对多个设备进行描述, 系统会用 Identifier
的值区分不同的设备.

Screen 段: X Window 的显示模式的选择就是记录在这个段中的. 如果是 RedHat
则会有所有四种 (Mono, VGA16, SVGA, Accelerate) X Server 的段, 如果
是 Slackware, 则可能会只有一个段, 就是对应安装系统的时候指定的显示
设备的.

下面以我使用的 accel X Server 说说各字段的涵义.

```
Section "Screen"
Driver "accel"
# 这个不用解释了吧, 是 Permedia II 显卡, 所以使用加速的驱动
# 这里的可能值是"vga2", "vga16", "svga", "accel", 根据自己
# 的实际情况设定. 这里跟 前面讲 X Server 时所说的 X 的连接
# 目标有关系, 我想这不难连接吧.
Device "3DLabs|Permedia II 2D+3D"
# 上面制定的显卡配制, 用 "3DLabs|Permedia II 2D+3D" 作为
# Identifier 的. 当然我可以使用自己喜欢的名字, 实际这里只是
# 个标记的作用
Monitor "Samsung SyncMaster 500s/500Ms"
# 这里的解释同 Device, 也是使用上面的 Identifier 的设定值
Subsection "Display"
# 显示子段, 可以有多个, 分别对应不同的色深, 即下面的 Depth
```

```
Depth 16
Modes "800x600"
# 所取的显示模式 这个值就是根据前面的 ModeLine 上面
# 的注释写的, 可以有多个, 用空格分开
# 这里需要说明的是, X Server 先检查这里的分辨率设置,
# 然后根据这个设置和上面标示的垂直和水平刷新频率决
# 定使用哪个 ModeLine, 如果在刷新频率的范围内有多种
# Mode 可以选择, 那么可以通过使用 Ctrl + '+'/'-'进
# 行动态切换.
ViewPort 0 0
# 这个是视图的左上角坐标
# 如果想用虚拟大小的屏幕, 可以在这里加上一句
#Virtual 1024 768
# 这样的话, 就可以使用大于当前实际屏幕的虚拟屏幕了,
# 屏幕会随这鼠标的移动而转换. 当然, 要保证有足够的
# 显存才行, 虚拟的屏幕大小要比实际的设定大..
EndSubsection
# 在这下面可以再加上 Depth 分别为 8 和 24 的子段, 格式和上
# 面一样.
EndSection
```

当然, 如果发现所选的设备驱动根实际的硬件设备不相符, 还可以重新安装驱动程序.如何在 Linux 下面安装软件, 我想就不再详细介绍了. 介绍这方面知识的资料很多.
<hr>

上一页 目录 下一页

上一页 目录 下一页

文件管理命令详解

which

功能说明：查找文件。

语　　法：which [文件...]

补充说明：which指令会在环境变量$PATH设置的目录里查找符合条件的文件。

参　　数：
　-n<文件名长度>　指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名。
　-p<文件名长度>　与-n参数相同，但此处的<文件名长度>包括了文件的路径。
　-w　指定输出时栏位的宽度。
　-V　显示版本信息。

----

whereis

功能说明：查找文件。

语　　法：whereis [-bfmsu][-B <目录>...][-M <目录>...][-S <目录>...][文件...]

补充说明：whereis指令会在特定目录中查找符合条件的文件。这些文件的烈性应属于原始代码，二进制文件，或是帮助文件。

参　　数：
　-b 　只查找二进制文件。
　-B<目录> 　只在设置的目录下查找二进制文件。
　-f 　不显示文件名前的路径名称。
　-m 　只查找说明文件。
　-M<目录> 　只在设置的目录下查找说明文件。
　-s 　只查找原始代码文件。
　-S<目录> 　只在设置的目录下查找原始代码文件。
　-u 　查找不包含指定类型的文件。

----

umask

功能说明：指定在建立文件时预设的权限掩码。

语　　法：umask [-S][权限掩码]

补充说明：umask可用来设定[权限掩码]。[权限掩码]是由3个八进制的数字所组成，将现有的存取权限减掉权限掩码后，即可产生建立文件时预设的权限。

参　　数：
-S 　以文字的方式来表示权限掩码。

----

touch

功能说明：改变文件或目录时间。

语　　法：touch [-acfm][-d <日期时间>][-r <参考文件或目录>][-t <日期时间>][--help]　　[--

version][文件或目录...] 或 touch [-acfm][--help][--version][日期时间][文件或目录...]

补充说明：使用touch指令可更改文件或目录的日期时间，包括存取时间和更改时间。

参　　数：
-a或--time=atime或--time=access或--time=use　只更改存取时间。
-c或--no-create　不建立任何文件。
-d<时间日期>　使用指定的日期时间，而非现在的时间。
-f　此参数将忽略不予处理，仅负责解决BSD版本touch指令的兼容性问题。
-m或--time=mtime或--time=modify　只更改变动时间。
-r<参考文件或目录>　把指定文件或目录的日期时间，统统设成和参考文件或目录的日期时间相同。

-t<日期时间>　使用指定的日期时间，而非现在的时间。
--help　在线帮助。
--version　显示版本信息。

----

tmpwatch(temporary watch)

功能说明：删除暂存文件。

语　　法：tmpwatch [-afqv][--test][超期时间][目录...]

补充说明：执行tmpwatch指令可删除不必要的暂存文件，您可以设置文件超期时间，单位以小时计算。

参　　数：
-a或--all　删除任何类型的文件。
-f或--force　强制删除文件或目录，其效果类似rm指令的"-f"参数。
-q或--quiet　不显示指令执行过程。
-v或--verbose　详细显示指令执行过程。
-test　仅作测试，并不真的删除文件或目录。

----

tee

功能说明：读取标准输入的数据，并将其内容输出成文件。

语　　法：tee [-ai][--help][--version][文件...]

补充说明：tee指令会从标准输入设备读取数据，将其内容输出到标准输出设备，同时保存成文件。

参　　数：
　-a或--append　附加到既有文件的后面，而非覆盖它.
　-i-i或--ignore-interrupts　忽略中断信号。
　--help　在线帮助。
　--version　显示版本信息。

----

split

功能说明：切割文件。

语　　法：split [--help][--version][-<行数>][-b <字节>][-C <字节>][-l <行数>][要切割的文件][输出文件名]

补充说明：split可将文件切成较小的文件，预设每1000行会切成一个小文件。

参　　数：
　-<行数>或-l<行数>　指定每多少行就要切成一个小文件。
　-b<字节>　指定每多少字就要切成一个小文件。
　-C<字节>　与-b参数类似，但切割时尽量维持每行的完整性。
　--help　显示帮助。
　--version　显示版本信息。
　[输出文件名]　设置切割后文件的前置文件名，split会自动在前置文件名后再加上编号。

----

slocate(secure locate)

功能说明：查找文件或目录。

语　　法：slocate [-u][--help][--version][-d <目录>][查找的文件]

补充说明：slocate本身具有一个数据库，里面存放了系统中文件与目录的相关信息。

参　　数：
　-d<目录>或--database=<目录>　指定数据库所在的目录。
　-u　更新slocate数据库。
　--help　显示帮助。
　--version　显示版本信息。

----

rm(remove)

功能说明：删除文件或目录。

语　　法：rm [-dfirv][--help][--version][文件或目录...]

补充说明：执行rm指令可删除文件或目录，如欲删除目录必须加上参数"-r"，否则预设仅会删除文件。
参　　数：
　-d或--directory　直接把欲删除的目录的硬连接数据删成0，删除该目录。
　-f或--force　强制删除文件或目录。
　-i或--interactive　删除既有文件或目录之前先询问用户。
　-r或-R或--recursive　递归处理，将指定目录下的所有文件及子目录一并处理。
　-v或--verbose　显示指令执行过程。
　--help　在线帮助。
　--version　显示版本信息。

----

rhmask

功能说明：产生与还原加密文件。

语　　法：rhmask [加密文件][输出文件] 或 rhmask [-d][加密文件][源文件][输出文件]

补充说明：执行rhmask指令可制作加密过的文件，方便用户在公开的网络上传输该文件，而不至于被任意盗用。

参　　数：
-d　产生加密过的文件。

----

rcp(remote copy)

功能说明：远端复制文件或目录。

语　　法：rcp [-pr][源文件或目录][目标文件或目录] 或 rcp [-pr][源文件或目录...][目标文件]
补充说明：rcp指令用在远端复制文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它灰把前面指定的所有文件或目录复制到该目录中。

参　　数：
-p　保留源文件或目录的属性，包括拥有者，所属群组，权限与时间。

-r　递归处理，将指定目录下的文件与子目录一并处理。

----

mdir

功能说明：显示MS-DOS目录。

语　　法：mdir [-afwx/][目录]

补充说明：mdir为mtools工具指令，模拟MS-DOS的dir指令，可显示MS-DOS文件系统中的目录内容。

参　　数：
-/ 显示目录下所有子目录与文件。
-a　显示隐藏文件。
-f　不显示磁盘所剩余的可用空间。
-w　仅显示目录或文件名称，并以横排方式呈现，以便一次能显示较多的目录或文件。
-X　仅显示目录下所有子目录与文件的完整路径，不显示其他信息。

----

mdel

功能说明：删除MS-DOS文件。

语　　法：mdel [文件...]

补充说明：mdel为mtools工具指令，模拟MS-DOS的del指令，可删除MS-DOS文件系统中的文件。

----

mcopy

功能说明：复制MS-DOS文件。

语　　法：mcopy [-bnmpQt/][源文件][目标文件或目录]

补充说明：mcopy为mtools工具指令，可在MS-DOS文件系统中复制文件或是在MS-DOS与Linux的文件系统之间复制文件。mcopy会MS-DOS磁盘驱动器代号出现的位置来判断如何复制文件。

参　　数：
-b　批处理模式。

-n　覆盖其他文件时，不需进行确认而直接覆盖。
-m　将源文件修改时间设为目标文件的修改时间。
-p　将源文件属性设为目标文件的属性。
-Q　复制多个文件时，若发生错误，则立即结束程序。
-t　转换文本文件。
-/　复制子目录以及其中的所有文件。

----

mc(midnight commander)

功能说明：提供一个菜单式的文件管理程序。

语　　法：mc [-abcdfhkPstuUVx][-C <参数>][-l <文件>][-v <文件>][目录]

补充说明：执行mc之后，将会看到菜单式的文件管理程序，共分成4个部分。

参　　数：
  -a　当mc程序画线时不用绘图字符画线。
  -b　使用单色模式显示。
  -c　使用彩色模式显示。
  -C<参数>　指定显示的颜色。
  -d　不使用鼠标。
  -f　显示mc函数库所在的目录。
  -h　显示帮助。
  -k　重设softkeys成预设置。
  -l<文件>　在指定文件中保存ftpfs对话窗的内容。
  -P　程序结束时，列出最后的工作目录。
  -s　用慢速的终端机模式显示，在这模式下将减少大量的绘图及文字显示。
  -t　使用TEMPCAP变量设置终端机，而不使用预设置。
  -u　不用目前的shell程序。
  -U　使用目前的shell程序。
  -v<文件>　使用mc的内部编辑器来显示指定的文件。
  -V　显示版本信息。
  -x　指定以xterm模式显示。

----

mattrib

功能说明：变更或显示MS-DOS文件的属性。

语　　法：mattrib [+/-][ahrs][-/X][文件] 或 mattrib [-/X][文件]

补充说明：mattrib为mtools工具指令，模拟MS-DOS的attrib指令，可变更MS-DOS文件的属性。

参　　数：
  +/-　"+"代表开启指定的文件属性。"-"代表关闭指定的文件属性。
  a　备份属性。使备份程序可用来判断文件是否已经备份过。
  h　隐藏属性。
  r　只读属性。
  s　系统属性。
  -/　列出指定目录及子目录下所有文件的属性。
  -X　显示文件属性时，中间不输入空格。

----

lsattr(list attribute)

功能说明：显示文件属性。

语　　法：lsattr [-adlRvV][文件或目录...]

补充说明：用chattr执行改变文件或目录的属性，可执行lsattr指令查询其属性。

参　　数：
-a　显示所有文件和目录，包括以"."为名称开头字符的额外内建，现行目录"."与上层目录".."。
-d　显示，目录名称，而非其内容。
-l　此参数目前没有任何作用。
-R　递归处理，将指定目录下的所有文件及子目录一并处理。
-v　显示文件或目录版本。
-V　显示版本信息。

----

locate

功能说明：查找文件。

语　　法：locate [-d <数据库文件>][--help][--version][范本样式...]

补充说明：locate指令用于查找符合条件的文件，它会去保存文件与目录名称的数据库内，查找合乎范本样式条件的文件或目录。

参　　数：
  -d<数据库文件>或--database=<数据库文件> 设置locate指令使用的数据库。locate指令预设的数据

库位于/var/lib/slocate目录里，文件名为slocate.db，您可使用这个参数另行指定。
　--help　在线帮助。
　--version　显示版本信息。

----

git(gnu interactive tools)

功能说明：文字模式下的文件管理员。

语　　法：git

补充说明：git是用来管理文件的程序，它十分类似DOS下的Norton Commander，具有互动式操作界面。它的操作方法和Norton Commander几乎一样，略诉如下：
F1 ：执行info指令，查询指令相关信息，会要求您输入欲查询的名称。
F2 ：执行cat指令，列出文件内容。
F3 ：执行gitview指令，观看文件内容。
F4 ：执行vi指令，编辑文件内容。
F5 ：执行cp指令，复制文件或目录，会要求您输入目标文件或目录。
F6 ：执行mv指令，移动文件或目录，或是更改其名称，会要求您输入目标文件或目录。
F7 ：执行mkdir指令，建立目录。
F8 ：执行rm指令，删除文件或目录。
F9 ：执行make指令，批处理执行指令或编译程序时，会要求您输入相关命令。
F10 ：离开git文件管理员。

od(octal dump)

功能说明：输出文件内容。

语　　法：od [-abcdfhilovx][-A <字码基数>][-j <字符数目>][-N <字符数目>][-s <字符串字符数>][-t <输出格式>][-w <每列字符数>][--help][--version][文件...]

补充说明：od指令会读取所给予的文件的内容，并将其内容以八进制字码呈现出来。

参　　数：
　-a　此参数的效果和同时指定"-ta"参数相同。
　-A<字码基数>　选择要以何种基数计算字码。
　-b　此参数的效果和同时指定"-toC"参数相同。
　-c　此参数的效果和同时指定"-tC"参数相同。
　-d　此参数的效果和同时指定"-tu2"参数相同。
　-f　此参数的效果和同时指定"-tfF"参数相同。
　-h　此参数的效果和同时指定"-tx2"参数相同。
　-i　此参数的效果和同时指定"-td2"参数相同。
　-j<字符数目>或--skip-bytes=<字符数目>　略过设置的字符数目。
　-l　此参数的效果和同时指定"-td4"参数相同。

-N<字符数目>或--read-bytes=<字符数目>　到设置的字符数目为止。
-o　此参数的效果和同时指定"-to2"参数相同。
-s<字符串字符数>或--strings=<字符串字符数>　只显示符合指定的字符数目的字符串。
-t<输出格式>或--format=<输出格式>　设置输出格式。
-v或--output-duplicates　输出时不省略重复的数据。
-w<每列字符数>或--width=<每列字符数>　设置每列的最大字符数。
-x　此参数的效果和同时指定"-h"参数相同。
--help　在线帮助。
--version　显示版本信息。

----

mv

功能说明：移动或更名现有的文件或目录。

语　　法：mv [-bfiuv][--help][--version][-S <附加字尾>][-V <方法>][源文件或目录][目标文件或目录]

补充说明：mv可移动文件或目录，或是更改文件或目录的名称。

参　　数：
-b或--backup　若需覆盖文件，则覆盖前先行备份。
-f或--force　若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文　件或目录。
-i或--interactive　覆盖前先行询问用户。
-S<附加字尾>或
--suffix=<附加字尾>　与-b参数一并使用，可指定备份文件的所要附加的字尾。
-u或--update　在移动或更改文件名时，若目标文件已存在，且其文件日期比源文件新，则不覆盖目标文件。
-v或--verbose　执行时显示详细的信息。
-V=<方法>或
--version-control=<方法>　与-b参数一并使用，可指定备份的方法。
--help　显示帮助。
--version　显示版本信息。

----

mtoolstest

功能说明：测试并显示mtools的相关设置。

语　　法：mtoolstest

补充说明：mtoolstest为mtools工具指令，可读取与分析mtools的配置文件，并在屏幕上显示结果。

----

mtools

功能说明：显示mtools支持的指令。

语　　法：mtools

补充说明：mtools为MS-DOS文件系统的工具程序，可模拟许多MS-DOS的指令。这些指令都是mtools的符号连接，因此会有一些共同的特性。

参　　数：
-a　长文件名重复时自动更改目标文件的长文件名。
-A　短文件名重复但长文件名不同时自动更改目标文件的短文件名。
-o　长文件名重复时，将目标文件覆盖现有的文件。
-O　短文件名重复但长文件名不同时，将目标文件覆盖现有的文件。
-r　长文件名重复时，要求用户更改目标文件的长文件名。
-R　短文件名重复但长文件名不同时，要求用户更改目标文件的短文件名。
-s　长文件名重复时，则不处理该目标文件。
-S　短文件名重复但长文件名不同时，则不处理该目标文件。
-v　执行时显示详细的说明。
-V　显示版本信息。

----

mshowfat

功能说明：显示MS-DOS文件在FAT中的记录。

语　　法：mshowfat [文件...]

补充说明：mshowfat为mtools工具指令，可显示MS-DOS文件在FAT中的记录编号。

----

mren

功能说明：更改MS-DOS文件或目录的名称，或是移动文件或目录。

语　　法：mren [源文件或目录...][目标文件或目录]

补充说明：mren为MS-DOS工具指令，模拟MS-DOS的ren指令，可更改MS-DOS文件或目录名称。除此之外，

ren也可移动文件或目录，但仅限于在同一设备内。

----

mread

功能说明：将MS-DOS文件复制到Linux/Unix的目录中。

语　　法：mread [MS-DOS文件...][Linux文件或目录]

补充说明：mread为mtools工具指令，可将MS-DOS文件复制到Linux的文件系统中。这个指令目前已经不常用，一般都使用mcopy指令来代替。

----

mmove

功能说明：在MS-DOS文件系统中，移动文件或目录，或更改名称。

语　　法：mmove [源文件或目录...][目标文件或目录]

补充说明：mmove为mtools工具指令，模拟MS-DOS的move指令，可在MS-DOS文件系统中移动现有的文件或目录，或是更改现有文件或目录的名称。

----

mktemp

功能说明：建立暂存文件。

语　　法：mktemp [-qu][文件名参数]

补充说明：mktemp可建立一个暂存文件，供shell script使用。

参　　数：

-q　执行时若发生错误，不会显示任何信息。

-u　暂存文件会在mktemp结束前先行删除。

[文件名参数]　文件名参数必须是以"自订名称.XXXXXX"的格式。

----

find

功能说明：查找文件或目录。

语　　法：find [目录...][-amin <分钟>][-anewer <参考文件或目录>][-atime <24小时数>][-cmin <分钟>][-cnewer <参考文件或目录>][-ctime <24小时数>][-daystart][-depyh][-empty][-exec <执行指令>][-false][-fls <列表文件>][-follow][-fprint <列表文件>][-fprint0 <列表文件>][-fprintf <列表文件><输出格式>][-fstype <文件系统类型>][-gid <群组识别码>][-group <群组名称>][-help][-ilname <范本样式>][-iname <范本样式>][-inum][-ipath <范本样式>][-iregex <范本样式>][-links <连接数目>][-lname <范本样式>][-ls][-maxdepth <目录层级>][-mindepth <目录层级>][-mmin <分钟>][-mount]
[-mtime <24小时数>][-name <范本样式>][-newer <参考文件或目录>][-nogroup][noleaf] [-nouser][-ok <执行指令>][-path <范本样式>][-perm <权限数值>][-print][-print0][-printf <输出格式>][-prune][-regex <范本样式>][-size <文件大小>][-true][-type <文件类型>][-uid <用户识别码>][-used <日数>][-user <拥有者名称>][-version][-xdev][-xtype <文件类型>]

补充说明：find指令用于查找符合条件的文件。任何位于参数之前的字符串都将被视为欲查找的目录。

参　　数：

-amin<分钟>　查找在指定时间曾被存取过的文件或目录，单位以分钟计算。
-anewer<参考文件或目录>　查找其存取时间较指定文件或目录的存取时间更接近现在的文件或目录。
-atime<24小时数>　查找在指定时间曾被存取过的文件或目录，单位以24小时计算。
-cmin<分钟>　查找在指定时间之时被更改的文件或目录。
-cnewer<参考文件或目录>　查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录。
-ctime<24小时数>　查找在指定时间之时被更改的文件或目录，单位以24小时计算。
-daystart　从本日开始计算时间。
-depth　从指定目录下最深层的子目录开始查找。
-expty　寻找文件大小为0 Byte的文件，或目录下没有任何子目录或文件的空目录。
-exec<执行指令>　假设find指令的回传值为True，就执行该指令。
-false　将find指令的回传值皆设为False。
-fls<列表文件>　此参数的效果和指定"-ls"参数类似，但会把结果保存为指定的列表文件。
-follow　排除符号连接。
-fprint<列表文件>　此参数的效果和指定"-print"参数类似，但会把结果保存成指定的列表文件。
-fprint0<列表文件>　此参数的效果和指定"-print0"参数类似，但会把结果保存成指定的列表文件。
-fprintf<列表文件><输出格式>　此参数的效果和指定"-printf"参数类似，但会把结果保存成指定的列表文件。
-fstype<文件系统类型>　只寻找该文件系统类型下的文件或目录。
-gid<群组识别码>　查找符合指定之群组识别码的文件或目录。
-group<群组名称>　查找符合指定之群组名称的文件或目录。
-help或--help　在线帮助。
-ilname<范本样式>　此参数的效果和指定"-lname"参数类似，但忽略字符大小写的差别。
-iname<范本样式>　此参数的效果和指定"-name"参数类似，但忽略字符大小写的差别。
-inum　查找符合指定的inode编号的文件或目录。
-ipath<范本样式>　此参数的效果和指定"-ipath"参数类似，但忽略字符大小写的差别。

-iregex<范本样式>　此参数的效果和指定"-regexe"参数类似，但忽略字符大小写的差别。
-links<连接数目>　查找符合指定的硬连接数目的文件或目录。
-iname<范本样式>　指定字符串作为寻找符号连接的范本样式。
-ls　假设find指令的回传值为True，就将文件或目录名称列出到标准输出。
-maxdepth<目录层级>　设置最大目录层级。
-mindepth<目录层级>　设置最小目录层级。
-mmin<分钟>　查找在指定时间曾被更改过的文件或目录，单位以分钟计算。
-mount　此参数的效果和指定"-xdev"相同。
-mtime<24小时数>　查找在指定时间曾被更改过的文件或目录，单位以24小时计算。
-name<范本样式>　指定字符串作为寻找文件或目录的范本样式。
-newer<参考文件或目录>　查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录。

-nogroup　找出不属于本地主机群组识别码的文件或目录。
-noleaf　不去考虑目录至少需拥有两个硬连接存在。
-nouser　找出不属于本地主机用户识别码的文件或目录。
-ok<执行指令>　此参数的效果和指定"-exec"参数类似，但在执行指令之前会先询问用户，若回答"y"或"Y"，则放弃执行指令。
-path<范本样式>　指定字符串作为寻找目录的范本样式。
-perm<权限数值>　查找符合指定的权限数值的文件或目录。
-print　假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称之前皆有"./"字符串。
-print0　假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式为全部的名称皆在同一行。
-printf<输出格式>　假设find指令的回传值为True，就将文件或目录名称列出到标准输出。格式可以自行指定。
-prune　不寻找字符串作为寻找文件或目录的范本样式。
-regex<范本样式>　指定字符串作为寻找文件或目录的范本样式。
-size<文件大小>　查找符合指定的文件大小的文件。
-true　将find指令的回传值皆设为True。
-typ<文件类型>　只寻找符合指定的文件类型的文件。
-uid<用户识别码>　查找符合指定的用户识别码的文件或目录。
-used<日数>　查找文件或目录被更改之后在指定时间曾被存取过的文件或目录，单位以日计算。
-user<拥有者名称>　查找符合指定的拥有者名称的文件或目录。
-version或--version　显示版本信息。
-xdev　将范围局限在先行的文件系统中。
-xtype<文件类型>　此参数的效果和指定"-type"参数类似，差别在于它针对符号连接检查。

----

file

功能说明：辨识文件类型。

语　法：file [-beLvz][-f <名称文件>][-m <魔法数字文件>...][文件或目录...]

补充说明：通过file指令，我们得以辨识该文件的类型。

参　　数：
　-b 　列出辨识结果时，不显示文件名称。
　-c 　详细显示指令执行过程，便于排错或分析程序执行的情形。
　-f<名称文件> 　指定名称文件，其内容有一个或多个文件名称呢感，让file依序辨识这些文件，格式为每列一个文件名称。
　-L 　直接显示符号连接所指向的文件的类别。
　-m<魔法数字文件> 　指定魔法数字文件。
　-v 　显示版本信息。
　-z 　尝试去解读压缩文件的内容。

----

diffstat(differential status)

功能说明：根据diff的比较结果，显示统计数字。

语　　法：diff [-wV][-n <文件名长度>][-p <文件名长度>]

补充说明：diffstat读取diff的输出结果，然后统计各文件的插入，删除，修改等差异计量。

参　　数：
　-n<文件名长度> 　指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名。
　-p<文件名长度> 　与-n参数相同，但此处的<文件名长度>包括了文件的路径。
　-w 　指定输出时栏位的宽度。
　-V 　显示版本信息。

----

indent

功能说明：调整C原始代码文件的格式。

语　　法：indent [参数][源文件] 或 indent [参数][源文件][-o 目标文件]

补充说明：indent可辨识C的原始代码文件，并加以格式化，以方便程序设计师阅读。

参　　数：
　-bad或--blank-lines-after-declarations 　在声明区段或加上空白行。
　-bap或--blank-lines-after-procedures 　在程序或加上空白行。
　-bbb或--blank-lines-after-block-comments 　在注释区段后加上空白行。
　-bc或--blank-lines-after-commas 　在声明区段中，若出现逗号即换行。
　-bl或--braces-after-if-line 　if(或是else,for等等)与后面执行区段的"{"不同行，且"}"自成一行。
　-bli<缩排格数>或--brace-indent<缩排格数> 　设置{ }缩排的格数。

-br或--braces-on-if-line　if(或是else,for等等)与后面执行跛段的"{"不同行，且"}"自成一行。

-bs或--blank-before-sizeof　在sizeof之后空一格。

-c<栏数>或--comment-indentation<栏数>　将注释置于程序码右侧指定的栏位。

-cd<栏数>或--declaration-comment-column<栏数>　将注释置于声明右侧指定的栏位。

-cdb或--comment-delimiters-on-blank-lines　注释符号自成一行。

-ce或--cuddle-else　将else置于"}"(if执行区段的结尾)之后。

-ci<缩排格数>或--continuation-indentation<缩排格数>　叙述过长而换行时，指定换行后缩排的格数。

-cli<缩排格数>或--case-indentation-<缩排格数>　使用case时，switch缩排的格数。

-cp<栏数>或-else-endif-column<栏数>　将注释置于else与elseif叙述右侧定的栏位。

-cs或--space-after-cast　在cast之后空一格。

-d<缩排格数>或-line-comments-indentation<缩排格数>　针对不是放在程序码右侧的注释，设置其缩排格数。

-di<栏数>或--declaration-indentation<栏数>　将声明区段的变量置于指定的栏位。

-fc1或--format-first-column-comments　针对放在每行最前端的注释，设置其格式。

-fca或--format-all-comments　设置所有注释的格式。

-gnu或--gnu-style　指定使用GNU的格式，此为预设值。

-i<格数>或--indent-level<格数>　设置缩排的格数。

-ip<格数>或--parameter-indentation<格数>　设置参数的缩排格数。

-kr或--k-and-r-style　指定使用Kernighan&Ritchie的格式。

-lp或--continue-at-parentheses　叙述过长而换行，且叙述中包含了括弧时，将括弧中的每行起始栏位内容垂直对其排列。

-nbad或--no-blank-lines-after-declarations　在声明区段后不要加上空白行。

-nbap或--no-blank-lines-after-procedures　在程序后不要加上空白行。

-nbbb或--no-blank-lines-after-block-comments　在注释区段后不要加上空白行。

-nbc或--no-blank-lines-after-commas　在声明区段中，即使出现逗号，仍旧不要换行。

-ncdb或--no-comment-delimiters-on-blank-lines　注释符号不要自成一行。

-nce或--dont-cuddle-else　不要将else置于"}"之后。

-ncs或--no-space-after-casts　不要在cast之后空一格。

-nfc1或--dont-format-first-column-comments　不要格式化放在每行最前端的注释。

-nfca或--dont-format-comments　不要格式化任何的注释。

-nip或--no-parameter-indentation　参数不要缩排。

-nlp或--dont-line-up-parentheses　叙述过长而换行，且叙述中包含了括弧时，不用将括弧中的每行起始栏位垂直对其排列。

-npcs或--no-space-after-function-call-names　在调用的函数名称之后，不要加上空格。

-npro或--ignore-profile　不要读取indent的配置文件.indent.pro。

-npsl或--dont-break-procedure-type　程序类型与程序名称放在同一行。

-nsc或--dont-star-comments　注解左侧不要加上星号(*)。

-nsob或--leave-optional-semicolon　不用处理多余的空白行。

-nss或--dont-space-special-semicolon　若for或while区段仅有一行时，在分号前不加上空格。

-nv或--no-verbosity　不显示详细的信息。

-orig或--original　使用Berkeley的格式。

-pcs或--space-after-procedure-calls　在调用的函数名称与"{"之间加上空格。

-psl或--procnames-start-lines　程序类型置于程序名称的前一行。

-sc或--start-left-side-of-comments　在每行注释左侧加上星号(*)。

-sob或--swallow-optional-blank-lines　删除多余的空白行。

-ss或--space-special-semicolon　若for或swile区段今有一行时，在分号前加上空格。
-st或--standard-output　将结果显示在标准输出设备。
-T　数据类型名称缩排。
-ts〈格数〉或--tab-size〈格数〉　设置tab的长度。
-v或--verbose　执行时显示详细的信息。
-version　显示版本信息。

----

paste

功能说明：合并文件的列。

语　法：paste [-s][-d 〈间隔字符〉][--help][--version][文件...]

补充说明：paste指令会把每个文件以列对列的方式，一列列地加以合并。

参　数：
-d〈间隔字符〉或--delimiters=〈间隔字符〉　用指定的间隔字符取代跳格字符。
-s或--serial　串列进行而非平行处理。
--help　在线帮助。
--version　显示帮助信息。

----

diff(differential)

功能说明：比较文件的差异。

语　法：diff [-abBcdefHilnNpPqrstTuvwy][-〈行数〉][-C 〈行数〉][-D 〈巨集名称〉][-I 〈字符或字符串〉][-S 〈文件〉][-W 〈宽度〉][-x 〈文件或目录〉][-X 〈文件〉][--help][--left-column][--suppress-common-line][文件或目录1][文件或目录2]

补充说明：diff以逐行的方式，比较文本文件的异同处。所是指定要比较目录，则diff会比较目录中相同文件名的文件，但不会比较其中子目录。

参　数：
-〈行数〉　指定要显示多少行的文本。此参数必须与-c或-u参数一并使用。
-a或--text　diff预设只会逐行比较文本文件。
-b或--ignore-space-change　不检查空格字符的不同。
-B或--ignore-blank-lines　不检查空白行。
-c　显示全部内文，并标出不同之处。
-C〈行数〉或--context〈行数〉　与执行"-c-〈行数〉"指令相同。

-d或--minimal　使用不同的演算法，以较小的单位来做比较。
-D<巨集名称>或ifdef<巨集名称>　此参数的输出格式可用于前置处理器巨集。
-e或--ed　此参数的输出格式可用于ed的script文件。
-f或-forward-ed　输出的格式类似ed的script文件，但按照原来文件的顺序来显示不同处。
-H或--speed-large-files　比较大文件时，可加快速度。
-I<字符或字符串>或--ignore-matching-lines<字符或字符串>　若两个文件在某几行有所不同，而这几行同时都包含了选项中指定的字符或字符串，则不显示这两个文件的差异。
-i或--ignore-case　不检查大小写的不同。
-l或--paginate　将结果交由pr程序来分页。
-n或--rcs　将比较结果以RCS的格式来显示。
-N或--new-file　在比较目录时，若文件A仅出现在某个目录中，预设会显示：
Only in目录：文件A若使用-N参数，则diff会将文件A与一个空白的文件比较。
-p　若比较的文件为C语言的程序码文件时，显示差异所在的函数名称。
-P或--unidirectional-new-file　与-N类似，但只有当第二个目录包含了一个第一个目录所没有的文件时，才会将这个文件与空白的文件做比较。
-q或--brief　仅显示有无差异，不显示详细的信息。
-r或--recursive　比较子目录中的文件。
-s或--report-identical-files　若没有发现任何差异，仍然显示信息。
-S<文件>或--starting-file<文件>　在比较目录时，从指定的文件开始比较。
-t或--expand-tabs　在输出时，将tab字符展开。
-T或--initial-tab　在每行前面加上tab字符以便对齐。
-u,-U<列数>或--unified=<列数>　以合并的方式来显示文件内容的不同。
-v或--version　显示版本信息。
-w或--ignore-all-space　忽略全部的空格字符。
-W<宽度>或--width<宽度>　在使用-y参数时，指定栏宽。
-x<文件名或目录>或--exclude<文件名或目录>　不比较选项中所指定的文件或目录。
-X<文件>或--exclude-from<文件>　您可以将文件或目录类型存成文本文件，然后在=<文件>中指定此文本文件。
-y或--side-by-side　以并列的方式显示文件的异同之处。
--help　显示帮助。
--left-column　在使用-y参数时，若两个文件某一行内容相同，则仅在左侧的栏位显示该行内容。
--suppress-common-lines　在使用-y参数时，仅显示不同之处。

----

ln(link)

功能说明：连接文件或目录。

语　法：ln [-bdfinsv][-S <字尾备份字符串>][-V <备份方式>][--help][--version][源文件或目录][目标文件或目录] 或 ln [-bdfinsv][-S <字尾备份字符串>][-V <备份方式>][--help][--version][源文件或目录...][目的目录]

补充说明：ln指令用在连接文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则会把前面指定的所有文件或目录复制到该目录中。若同时指定多个文件或目录，且最

后的目的地并非是一个已存在的目录，则会出现错误信息。

参　　数：

-b或--backup　删除，覆盖目标文件之前的备份。
-d或-F或--directory　建立目录的硬连接。
-f或--force　强行建立文件或目录的连接，不论文件或目录是否存在。
-i或--interactive　覆盖既有文件之前先询问用户。
-n或--no-dereference　把符号连接的目的目录视为一般文件。
-s或--symbolic　对源文件建立符号连接，而非硬连接。
-S<字尾备份字符串>或--suffix=<字尾备份字符串>　用"-b"参数备份目标文件.

系统安装FAQ和问题索引

1.什么是linux?
简单的说就是linux kernel
2.redhat linux，mandrake linux，debian linux，suse linux等等各种linux有何不同?
就象不同牌子的，不同味道的雪糕。牌子不一样，味道不一样，但都是雪糕。
3.我该用哪个linux?
新手可以用redhat或mandrake的，redhat用的人相对较多，所以我推荐用redhat，比如redhat7.3或redhat8.0
4.如何获得linux发行版?

5.如何安装linux?
Redhat8的安装教程，安装异曲同工会了这个，其他也好说
http://www.linuxfans.org/nuke/modules.php?name=Forums&file=viewtopic&t=2966

6.linux分区是怎么回事?

请参考
http://www.linuxfans.org/nuke/modules.php?name=Forums&file=viewtopic&t=10843
7.gnome是什么? kde又是什么?

gnome和kde都是当前linux系统比较常用并且界面比较友好的桌面环境。

8.grub是什么? lilo又是什么东东?

lilo和grub都是一个在系统启动时运行的程序，它允许选择用于引导计算机的操作系统。可以用lilo或

grub引导几个不同的操作系统，如：Linux和MS-DOS、Windows 98、win2000、winXP等

9.安装时把grub(lilo)写到linux分区的引导区还是主引导扇区(MBR)？
如果你想电脑一启动就直接进入操作系统启动菜单就把grub(lilo)写到MBR上，如果写到linux分区的引导区则要用引导盘引导，建议写到MBR，方便点，至于说写到MBR不安全，该怎么解释呢？每装一次win98，MBR都会被修改一次，大家觉得有什么不安全的吗？
10.如何让多系统共存？
98系统的话用lilo(gurb)即可
os loader引导多系统
http://www.linuxfans.org/nuke/modules.php?name=Forums&file=viewtopic&t=1346

11.如何让linux启动后自动进入图形界面或不让它进入图形界面？
a.进入图形界面：安装时选图形启动方式或把/etc/inittab的id:x:initdefault:这行改成id:5:initdefault:。
b.不进入图形界面：安装时选文本启动方式或把/etc/inittab的id:x:initdefault:这行改成id:3:initdefault:。
12.如何在图形界面和控制台（字符界面）之间来回切换？
a.图形界面到控制台：Ctr+Alt+Fn(n=1,2,3,4,5,6)。
b.各控制台之间切换：Alt+Fn(n=1,2,3,4,5,6)。
c.控制台到图形：Alt+F7
13.如何在gnome和kde之间切换。
如果你是以图形登录方式登录linux，那么点击登录界面上的session（任务）即可以选择gnome和kde。
如果你是以文本方式登录，那执行switchdesk gnome或switchdesk kde，然后再startx就可以进入gnome或kde。
14.linux常用的命令有哪些？
ls：列目录。
用法：ls或ls dirName，参数：-a显示所有文件，-l详悉列出文件。
mkdir：建目录。
用法：mkdir dirName，参数：-p建多级目录，如：mkdir a/b/c/d/e/f -p
mount：挂载分区或镜像文件(.iso,.img)文件。
用法：
a.磁盘分区：mount deviceName mountPoint -o options，其中deviceName是磁盘分区的设备名，比如/dev/hda1,/dev/cdrom,/dev/fd0，mountPoint是挂载点，它是一个目录，options是参数，如果分区是linux分区，一般不用-o options，如果是windows分区那options可以是iocharset=cp936，这样windows分区里的中文文件名就能显示出来了。用例：比如/dev/hda5是linux分区，我要把它挂到目录a上（如没目录a那就先mkdir a），mount /dev/hda5 a，这样目录a里的东西就是分区hda5里的东西了，比如hda1是windows分区，要把它挂到b上，mount /dev/hda1 b -o iocharset=cp936。
b.镜像文件：mount fileName mountPoint -o loop，fileName是镜像文件名(*.iso,*.img)，其它的不用说了，跟上面一样。用例：如我有一个a.iso光盘镜像文件，mount a.iso a -o loop，这样进入目录a你就能浏览a.iso的内容了，*.img文件的用法一样。
find：查找文件。
用法：find inDir -name filename，inDir是你要在哪个目录找，filename是你要找的文件名(可以用通配符)，用通配符时filename做好用单引号引起来，否则有时会出错，用例：find . -name test*，在当前目录查找以test开头的文件。
grep：在文件里查找指定的字符串。
用法：grep string filename，在filename(可用通配符)里查找string(最好用双引号引起来)。参数：-r在所有子目录里的filename里找。用例：grep hello *.c -r在当前目录下（包括子目录）的所有.c文

件里查找hello。
vi：编辑器。不用说，用linux的话，这个东西一定要会用。
用法：（只能简单说一下），vi filename。filename就是你要编辑的文本文件。用了执行vi filename后，你可能会发现你无法编辑文本内容，不要着急，这是因为vi还没进入编辑状态，按a或i就可以进入编辑状态了，进入编辑状态后你就可以编辑文本了。要退出编辑状态按Esc键就可以了。以下操作均要在非编辑状态下。查找文本：输入/和你要查找的文本并回车。退出：输入:和q并回车，如果你修改了文本，那么你要用:q!回车才能退出。保存：输入:w回车，如果是只读文件要用:w!。保存退出：输入:wq回车，如果是只读就:wq!回车。取消：按u就可以了，按一次就取消一步，可按多次取消多步。复制粘贴一行文本：把光标移到要复制的行上的任何地方，按yy（就是连按两次y），把光标移到要粘贴地方的上一行，按p，刚才那行文本就会被插入到光标所在行的下一行，原来光标所在行后面所有行会自动下移一行。复制粘贴多行文本：跟复制一行差不多，只是yy改成先输入要复制的行数紧接着按yy，后面的操作一样。把光标移到指定行：输入:和行号并回车，比如移到123行:123回车，移到结尾:$回车。
15..tar，.tar.gz，.bz2，.tar.bz2，.bz，.gz是什么文件，如何解开他们？
他们都是文件(压缩)包。
.tar：把文件打包，不压缩：tar cvf *.tar dirName 解开：tar xvf *.tar
.tar.gz：把文件打包并压缩：tar czvf *.tar.gz dirName 解开：tar xzvf *.tar.gz
.bz2：解开：bzip2 -d *.bz2
.bz：解开：bzip -d *.bz
.gz：解开：gzip -d *.gz
16.linux下如何解开.zip，.rar压缩文件？
rh8下有一个图形界面的软件file-roller可以做这件事。令外可以用unzip *.zip解开zip文件，unrar *.rar解开rar文件，不过unrar一般系统不自带，要到网上下载。
17.linux下如何浏览.iso光盘镜像文件？
a.建一个目录，如：mkdir a
b.把iso文件挂载到该目录上：mount -o loop xxxx.iso a
现在目录a里的内容就是iso文件里的内容了。
18.linux下如何访问其它分区？包括linux 分区和windows分区。
用mount命令，详见mount用法。
19.linux下如何配置显卡和显示器？

20.linux下如何配置网络？
用netconfig。“IP address:”就是要配置的IP地址，“Netmask:”子网掩码，“Default gateway (IP):”网关，“Primary nameserver:”DNS服务器IP。
21.如何让鼠标支持滚轮？
在配置鼠标时，选择微软的鼠标，并正确选择端口如ps2,usb等
22.linux下的互联网浏览器有哪些？
mozilla，netscape，opera，galeon(使用mozilla引擎)，phoniex(使用mozilla引擎)。常用的就这些吧。
23.linux的系统目录结构是怎么样的？

24.如何让控制台支持中文显示？
安装zhcon。zhcon需要libimm_server.so和libpth.so.13这两个库支持。一般的中文输入法应该都有libimm_server.so。libpth.so.13出自pth-1.3.x。把这两个文件放到/usr/lib下就行了。
25.如何配置grub？
修改/boot/grub/grub.conf文件。其中
“default=n”(n是个数字)是grub引导菜单默认被选中的项，n从0开始，0表示第一项，1表示第二项，依此类推。

“timeout=x”(x是一个数)是超时时间，单位是妙。也就是引导菜单显示后，如果x秒内用户不进行选择，那么grub将启动默认项。
“splashimage =xxxxxx”，这是引导菜单的背景图，先不理他。
其它常用项我用下面的例子来说明：
title Red Hat 8.0
root (hd1,6)
kernel /boot/vmlinuz-2.4.18-14 ro root=/dev/hdb7
initrd /boot/initrd-2.4.18-14.img
其中"Red Hat 8.0"是在启动菜单列表里显示的名字
root (hdx,y)用来指定你的boot分区位置，如果你没有分boot分区（本例就没分boot分区），那就指向根分区就行了，hdx是linux所在硬盘，hd0是第一块硬盘，hd1是第二块，依此类推。y是分区位置，从0开始，也就是等于分区号减一，比如你要指向的分区是hdx7，那么y就是6，如果是hdx1，那y就是0。注意root后面要有一个空格。
kernel /boot/vmlinuz-2.4.18-14，其中"/boot/vmlinuz-2.4.18-14"是你要用的内核路径，如果你编译了心内核，把它改成你的新内核的路径就行了。
ro就不用管，写上不会有错。
root=/dev/hdxx指定根分区，本例是hdb7，所以root=/dev/hdb7
initrd xxxxxxxxxxxxx这行不要也行，目前我还不清楚它是做什么用的。
上面是linux的，下面是windows的
title windows 98
rootnoverify (hd0,0)
chainloader +1
title xxxxxxx不用解释了，上面有解释。
rootnoverify (hdx,y)用来指定windows所在分区，x，y跟上面一样，注意rootnoverify后有空格。
chainloader +1照抄就行，注意空格。

26.如何配置lilo?
修改/etc/lilo.conf文件，然后运行lilo命令。
27.如何让linux启动后自动挂载制定分区？
编辑/etc/fstab文件，在里面加入相应的行，如果你对/etc/fstab的配置不熟悉，你可以用ptm，一个自动配置fstab的软件，在公社里应该有下载。
28.如何定制linux启动时启动的服务？
在控制台或终端下运行ntsysv，然后把要启动的服务选上，如果要开机不启动某个那就把他的选中去掉。
29.没有软驱如何修复grub/lilo引导菜单？
a.进入linux的安装界面。
b.按Ctrl+Alt+F2切换到控制台。
c.mkdir a建立目录a
d.mount /dev/hdax a把linux根分区mount到目录a上（hdax是你的根分区）
e.chroot a
f.执行/sbin/grub-install /dev/hdx(x=a,b,c,d) 或/sbin/lilo即可以重写引导。
g.exit退出chroot，umount a把根分区umount掉，rm a -rf 把目录a删除，Ctrl+Alt+Del重起
另外还可以参考：http://www.linuxfans.org/nuke/modules.php?name=Forums&file=viewtopic&t=5101

30.如何配置鼠标？
打开一个终端或进入控制台，运行mouseconfig命令即可配置鼠标！
31.如何从dos下进入linux?

首先你需要loadlin.exe这个程序，linux第一张安装盘里的dosutils目录里有，另外你需要的就是linux内核，/boot/vmlinuz-2.4.xx-xx就是，把他弄到windows盘中，并把他重命名为vmlinuz（主要是vmlinuz-2.4.xx-xx在dos下显示有点问题），另外dosutils目录里的autoboot里也有一个vmlinuz，也是linux内核，凑合能用，但很多设备不支持，如果linux引导程序或linux启动配置有问题，进不了linux，可以用这个内核进去修复linux。
然后用loadlin vmlinuz root=/dev/hdxx （hdxx是你的linux根分区），另外用loadlin vmlinuz root=/dev/hdxx single还可以实现单用户模式。
32.root密码丢失解决
GRUB:在引导装载程序菜单上，键入 [e] 来进入编辑模式。
你会面对一个引导项目列表。查找其中类似以下输出 的句行：
kernel /vmlinuz-2.4.18-0.4 ro root=/dev/hda2
按箭头键直到这一行被突出显示，然后按 [e]。
你现在可在文本结尾处空一格再添加 single 来告诉 GRUB 引导单用户 Linux 模式。按 [Enter] 键来使编辑结果生效。
你会被带会编辑模式屏幕，从这里，按 [b]， GRUB 就会引导单用户 Linux 模式。 结束载入后，你会面对一个与以下相似的 shell 提示：
sh-2.05#
现在，你便可以改变根命令，键入：
bash# passwd root
你会被要求重新键入口令来校验。结束后， 口令就会被改变，你便可以在提示下键入 reboot 来重新引导；然后，象平常一样登录为根用户。
LILO:当系统启动到出现LILO引导画面时，对于图形引导方式按TAB键进入文本方式，然后在LILO处输入linux single回车即可进入免密码的控制台，进入以后使用passwd命令修改root的密码即可

如何即时修改屏幕刷新频率和分辨率？
在XFree86 4.3（Redhat 9.0）后，有一个命令可以即时修改屏幕的刷新频率和分辨率，这个命令就是xrandr，使用方法：
该命令要在X下执行。
先执行xrandr看看你的显示器支持什么样的分辨率和刷新频率，然后用
xrandr -s <width>x<height> -r <rate>
如xrandr -s 1024x768 -r 84就是把分辨率设成1024x768，刷新频率是84（跟windows不太一样，windows可能是85）。使用前记得用xrandr看看都支持什么样的刷新频率，不要自己乱定。

如何修改控制台的分辨率？

对于喜欢在控制台下读写代码的朋友来说，linux默认的分辨率可能会显得太低，一个屏幕显示不了几行代码，如果把分辨率调高点，看起来会爽点。
grub：
编辑/boot/grub/grub.conf文件，在kernel /boot/vmlinuzxxxxx xxxxxxxxxxxx这行后面加上vga=0x31?也就是
kernel /boot/vmlinuzxxxxx xxxxxxxxxxxx vga=0x31?
lilo:
好像添加一行vga=0x31?就行了（很久没用lilo了，有点记不起来啦）
其中0x31?是设定具体分辨率值，可参考一下列表，不要乱设
1280x1024是0x31a

1024x768是0x317

800x600是0x314

640x480是0x311

一句话问答--linux常见问题

一句话问答(网络无关篇+网络相关篇)

----------------------------网络无关篇--------------------------

0001

修改主机名

vi /etc/sysconfig/network，修改HOSTNAME一行为HOSTNAME=主机名，重启后也能生效

0002

Ret Hat Linux启动到文字界面(不启动xwindow)

将/etc/inittab中 id:5:initdefault: 一行中的5改为3

0003

redhat的自动问题

在http://www.redhat.com/corp/support/errata/

具up2date，它能够测定哪些rpm包需要升级，然后自动从redhat的站点下载并完成安装。

0004

windows下看linux分区的软件

Paragon.Ext2FS.Anywhere.2.5.rar和explore2fs-1.00-pre4.zip

0005

mount用法

fat32的分区 mount -o codepage=936,iocharset=cp936 /dev/hda7 /mnt/cdrom

iso文件 mount -o loop /abc.iso /mnt/cdrom

软盘 mount /dev/fd0 /mnt/floppy

USB闪存 mount /dev/sda1 /mnt/cdrom

所有/etc/fstab内容 mount -a

可以指定文件格式"-t 格式", 格式可以为vfat, ext2, ext3等.

0006

在vmware的LINUX中使用本地硬盘的FAT分区

将本地的FAT分区共享，然后在VMWARE中使用SMBFS挂上。可以将如下的行放到/etc/fstab中：
//win_ip/D$ /mnt/d smbfs
defaults,auto,username=win_name,password=win_pass,codepage=936,iocharest=gb2312 0 0
其中win_ip是你的windows的IP地址；
D$是你的windows里面共享的D盘的共享名；
/mnt/d是要将该分区mount到linux的目录；
win_name和win_pass是你的WINDOWS里面可以读取该分区的用户，比如你的管理员名和密码。
如果你运行了/etc/rc.d/init.d/netfs，那么在启动的时候就会自动挂载这个分区。

0007.a
删除名为-a的文件
rm ./-a
rm -- -a 告诉rm这是最后一个选项，参见getopt
ls -i 列出inum，然后用find . -inum inum_of_thisfile -exec rm '{}' \;

0007.b
删除名为\a的文件
rm \\a

0007.c
删除名字带的/和‘\0'文件
这些字符是正常文件系统所不允许的字符，但可能在文件名中产生，如unix下的NFS文件系统在Mac系统上使用
1.解决的方法，把NFS文件系统在挂到不过滤'/'字符的系统下删除含特殊文件名的文件。
2.也可，将错误文件名的目录其它文件移走，ls -id 显示含该文件目录的inum，umount 文件系统，clri清除该目录的inum，fsck，mount，check your lost+found，rename the file in it.
最好是通过WINDOWS FTP过去就可以删除任何文件名的文件了!

0007.d
删除名字带不可见字符的文件
列出文件名并转储到文件：ls -l >aaa
然后编辑文件的内容加入rm命令使其内容成为删除上述文件的格式：
vi aaa
[rm -r *******
]
把文件加上执行权限 chmod +x aaa
执行 $aaa

0007.e
删除文件大小为零的文件
rm -i `find ./ -size 0`
find ./ -size 0 -exec rm {} \;
find ./ -size |xargs rm -f &非常有效
for file in * #自己定义需要删除的文件类型
do
if [! -s ${file}]
then

rm ${file}
echo "rm $file Success!"
fi
done

0008
redhat设置滚轮鼠标
进入X后，选择鼠标的配置，选择wheel mouse(ps/2)就可以了

0009
加装xwindow
用linux光盘启动，选择升级，然后单独选择包，安装即可

0010
删除linux分区
做一张partition magic的启动软盘,启动后删除. 或者用win2000的启动光盘启动,然后删除.

0011
新编译生成的gcc ，使用的标准连接库都在/usr/local/lib 下了，但使用的缺省的连接路径是 /usr/lib 怎样添加? （除了在每次编译时 增加 -L /usr/local/lib 以外)(sakulagi)
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
写到~/.bash_profile里面。

0012
不编译内核，mount ntfs分区
原装rh8，未升级或编译内核
1. 上google.com搜索并下载 kernel-ntfs-2.4.18-14.i686.rpm
2. rpm -ivh kernel-ntfs-2.4.18-14.i686.rpm
3. mkdir /mnt/c
4. mount -t ntfs /dev/hda1 /mnt/c

0013
redhat 8.0中用xmms听mp3
下载http://www.gurulabs.com/files/xmms-mp3-1.2.7-13.p.i386.rpm
rpm -Uvh xmms-mp3-1.2.7-13.p.i386.rpm

0014
找回忘记了的root口令（lilo/grub）
三种办法：
1.在系统进入单用户状态，直接用passwd root去更改
2.用安装光盘引导系统，进行linux rescue状态，将原来/分区挂接上来,作法如下：
cd /mnt
mkdir hd
mount -t auto /dev/hdaX(原来/分区所在的分区号） hd
cd hd
chroot ./
passwd root

这样可以搞定
3.将本机的硬盘拿下来，挂到其他的linux系统上，采用的办法与第二种相同
rh8中
一. lilo
1. 在出现 lilo: 提示时键入 linux single
画面显示 lilo: linux single
2. 回车可直接进入linux命令行
3. #vi /etc/shadow
将第一行，即以root开头的一行中root:后和下一个:前的内容删除，
第一行将类似于
root::......
保存
4. #reboot重启，root密码为空
二. grub
1. 在出现grub画面时，用上下键选中你平时启动linux的那一项(别选dos哟)，然后按e键
2. 再次用上下键选中你平时启动linux的那一项(类似于kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/)，然后按e键
3. 修改你现在见到的命令行，加入single，结果如下：
kernel /boot/vmlinuz-2.4.18-14 single ro root=LABEL=/
4. 回车返回，然后按b键启动，即可直接进入linux命令行
5. #vi /etc/shadow
将第一行，即以root开头的一行中root:后和下一个:前的内容删除，
第一行将类似于
root::......
保存
6. #reboot重启，root密码为空

0015
使ctrl + alt + del失效
vi /etc/inittab
将ca::ctrlaltdel:/sbin/shutdown -t3 -r now这行注释掉，就可以了

0016
gcc abc.c得到的a.out不能运行
./a.out

0017
文件在哪个rpm中(无双)
上http://www.rpmfind.net-qf 文件名得到

0018
把man或info的信息存为文本文件
以 tcsh 为例：
man tcsh | col -b > tcsh.txt
info tcsh -o tcsh.txt -s

0019

利用现存两个文件，生成一个新的文件
1. 取出两个文件的并集(重复的行只保留一份)
2. 取出两个文件的交集(只留下同时存在于两个文件中的文件)
3. 删除交集，留下其他的行
1. cat file1 file2 | sort | uniq
2. cat file1 file2 | sort | uniq -d
3. cat file1 file2 | sort | uniq -u

0020
设置com1口，让超级终端通过com1口进行登录
确认有/sbin/agetty，编辑/etc/inittab，添加
7:2345:respawn:/sbin/agetty /dev/ttyS0 9600
9600bps是因为联路由器缺省一般都是这种速率，也可以设成
19200、38400、57600、115200
修改/etc/securetty，添加一行：ttyS0，确保root用户能登录
重启机器，就可以拔掉鼠标键盘显示器（启动时最好还是要看看输出信息）了

0021
删除目录下所有文件包括子目录
rm -rf 目录名

0022
查看系统信息
cat /proc/cpuinfo - CPU (i.e. vendor, Mhz, flags like mmx)
cat /proc/interrupts - 中断
cat /proc/ioports - 设备IO端口
cat /proc/meminfo - 内存信息(i.e. mem used, free, swap size)
cat /proc/partitions - 所有设备的所有分区
cat /proc/pci - PCI设备的信息
cat /proc/swaps - 所有Swap分区的信息
cat /proc/version - Linux的版本号 相当于 uname -r

0023
去掉多余的回车符
sed 's/^M//' test.sh > back.sh， 注意^M是敲ctrl_v ctrl-m得到的
或者 dos2unix filename

0024
切换X桌面(lnx3000)
switchdesk KDE或者switchdesk GNOME

0025
通用的声卡驱动程序(lnx3000)
OSS http://www.opensound.com/ALSA www.alsa-project.org/

0026
改变redhat的系统语言/字符集(beming)

修改 /etc/sysconfig/i18n 文件，如
LAGN="en_US"，xwindow会显示英文界面，
LAGN="zh_CN.GB18030"，xwindow会显示中文界面。

0027
把屏幕设置为90列
stty cols 90

0028
使用md5sum文件
md5sum isofile > hashfile, 将 md5sum 档案与 hashfile 档案内容比对, 验证杂凑值
是否一致 md5sum –c hashfile

0029
一次解压多个zip文件
unzip "*"，注意引号不能少

0030
看pdf文件
使用xpdf或者安装acrobat reader for linux

0031
查找权限位为S的文件
find . -type f \(-perm -04000 -o -perm -02000 \) -exec ls -lg {} \;

0032
装中文输入法
以redhat8为例,xwindow及其终端下的不用说了吧,缺省就安装了,用ctrl-space呼出。
现在讨论纯console,请下载zhcon.gnuchina.org/download/src/zhcon-0.2.1.tar.gz,
放在任一目录中,tar xvfz zhcon-0.2.1.tar.gz, cd zhcon-0.2.1, ./configure,
make, make install。安装结束，要想使用，请运行zhcon，想退出，运行exit。

0033
把弹出的光盘收回来(beike)
#eject –t

----------------------------网络相关篇--------------------------
0001
让apache的默认字符集变为中文
找到 AddDefaultCharset ISO8859-1 一行，改为AddDefaultCharset GB2312

0002
永久更改ip
ifconfig eth0 新ip
然后编辑/etc/sysconfig/network-scripts/ifcfg-eth0，修改ip

0003

从Linux上远程显示Windows桌面(lnx3000)
安装rdesktop包

0004
手动添加默认网关
以root用户，执行: route add default gw 网关的IP
想更改网关
1 vi /etc/sysconfig/network-scripts/ifcfg-eth0
更改GATEWAY
2 /etc/init.d/network restart

0005
redhat 8.0上msn和qq
下载Gaim 0.58版：
http://www.linuxsir.org/gaim/gaim-0.58-2.i386.rpm
下载QQ插件 for gcc2.9版：
http://www.linuxsir.org/gaim/libqq-0.0.3-ft-0.58-gcc296.so.gz
将下载的文件放入/temp目录，然后将系统中已有的Gaim删除，即在终端仿真器中键入命令：rpm -e gaim。
开始安装
打开终端仿真器，继续执行下列命令安装Gaim 0.58版，即：
cd /temp （进入temp目录）
rpm -ivh gaim-0.58-2.i386.rpm （安装软件）
当安装成功后，你就可以在GNOME或KDE桌面建立Gaim图标了。
继续安装QQ插件，即键入命令：
gunzip libqq-0.0.3-ft-0.58-gcc296.so.gz (解压缩文件)
cp libqq-0.0.3-ft-0.58-gcc296.so /usr/lib/gaim （复制插件到gaim库目录中）
软件设置
首次启动Gaim 0.85版时，会出现的登录界面。先选择“插件”，在插件对话框中点击“加载”，分别将

libmsn.so和libqq-0.0.3-ft-0.58-gcc296.so文件装入，确认后关闭。然后再选择“所有帐号”，在出现的帐号编辑器中继续点击“增加”，当出现的修改帐号页面时，我们就可以输入自己的QQ或MSN号了，登录名填写QQ号码或MSN邮箱，密码填写对应的QQ或MSN密码，Alias填写自己的昵称，协议选择相应的QQ或MSN，其他的设置按默认的即可。当全部设置完成后就可以登录使用了。

0006
查出22端口现在运行什么程序
lsof -i :22

0007
查看本机的IP，gateway, dns
IP:
以root用户登录，执行ifconfig。其中eth0是第一块网卡，lo是默认的设备
Gateway:
以root用户登录，执行netstat -rn，以0.0.0.0开头的一行的Gateway即为默认网关
也可以查看/etc/sysconfig/network文件，里面有指定的地址！
DNS:

more /etc/reslov.conf，内容指定如下：
nameserver 202.96.69.38
nameserver 202.96.64.38

0008
RH8.0命令行下轻松改变ping 的TTL值(cgweb)
#sysctl -w net.ipv4.ip_default_ttl=N
(N=0~255),若N>255,则ttl=0

0009
RH8.0命令行下轻松改变系统配置默认值(houaq)
编辑/etc/sysctl.conf, 例如，将
net.ipv4.ip_forward = 0
变为
net.ipv4.ip_forward = 1
重启后生效，用sysctl -a查看可知

0010
mount局域网上其他windows机器共享出的目录
mount -t smbfs -o uname=guest,password=guest //machine/path /mnt/cdrom

0011
允许|禁止root通过SSH登陆(Fun-FreeBSD)
修改sshd_config:PermitRootLogin no|yes

0012
让root直接telnet登陆
编辑/etc/pam.d/login，去掉
auth required /lib/security/pam_securetty.so 这句话

0013
在linux接adsl设备
需要一个运转正常的Linux + 至少一块网卡 + 宽带设备已经申请完毕，同时已经开通
前市场上大概有几种ADSL设备，他们工作的方式有一些细微的差别。
就是通过虚拟拨号来完成上网的这一过程，也就是利用pppoe设备来进行虚拟拨号的
叫作全向猫，就是一种加电后自动的进行拨号的工作，然后留给我们的接口是RJ45,
大连地区一般留给我们的网关都是10.0.0.2,这种设备最容易对付，最后是直接分配
给用户一个固定的IP，相对大家来说也比较容易对付
1.第一种需要进行拨号：
这几种设备都是通过eth接口与计算机进行通讯的，所以先将硬件设备的连接作好，尤其是宽带猫的，一定要确认无误（否则一会儿要不去可不算我的事情）
然后启动系统，确认系统上是否安装rp-pppoe这个软件（通过rpm -qa|grep pppoe来查找），如没有安装的用户，在光盘里或是到网上去down一个来，安装上后，以root用户执行adsl-setup，这样就进入了adsl的资料的设定状态，要求输入申请宽带的用户名以及其他一些信息，确认没有问题，接受直至最后（里面都是E文，但是一看即能懂，比较简单，有关一个防火墙的设置，我一般都不用，选0，大家可以具体考虑）。
配置完成后，以root用户执行adsl-start，这样将进行adsl的拨号工作，正常就会一下

上线，如有什么具体问题，去看一下日志（/var/log/messages）里面告诉你什么了。
停掉adsl，执行adsl-stop就可以了（很简单的）
2.另外两种比较容易对付：
全向猫：只要将你的网卡的IP设置成一个10网段的IP，然后网关指到全向猫的IP
上（10.0.0.2),基本上不有太大的问题
 固定IP：就像配置本地儿的网卡一样，将IP，网关，DNS都按申请来的填写上就可以搞定了

<hr>
>

上一页 目录 下一页

上一页 目录 下一页

在Linux中共享Windows系统资源

在一部电脑上装有Linux和Windows的用户，可能需要在Linux中用到包括软盘、光盘以及Windows分区中的文件，如当你在Windows操作系统中上网，却下载了Linux的应用软件的时候。下面我就向大家详细介绍这方面的知识和具体的应用。

一、装载（mount）

（一）准备知识

在Linux系统中想要使用软驱的话，要先把它们装载到系统中，装载指令mount的格式如下：

mount －t 文件系统类型 设备文件名 装载目录

1. 文件系统类型

文件系统类型一般来说就是分区格式，依操作系统的不同而不同。下面将Linux系统支持的文件系统类型择要分列如表一：

2. 设备文件名

在Linux系统中，各个驱动器设备的命名和Windows中的规则（如A：、C：等）有很大的区别。所有的系统硬件设备都可以在/dev目录下找到对应的设备文件名。例如/dev/mouse就用来表示系统中的鼠标。磁盘驱动器的各个不同分区所对应的设备文件名列表如表二：

假设第一个IDE硬盘驱动器被分成数个分区，通常，第一个分区可以肯定其设备文件名为/dev/hda1，但其它分区的设备文件名就不是可以依次类推得到的（特别是当用户使用了诸如PTM之类的磁盘分区工具的时候）。如果你想要知道硬盘各分区所对应的设备文件名，可以在控制台下执行cfdisk的指令，则各分区对应的设备文件名就可以一目了然了。

3. 装载目录

通常我们都会在/mnt目录下面为需要装载的磁盘驱动器创建一个目录，不过这并不意味着它们不可以被装载于其它未被使用的目录中。再者，所谓装载目录，并不是将被装载的磁盘驱动器整个复制到本地，而仅仅是在本地提供一个装载点用以联系其它需要装载的磁盘驱动器。

Linux系统对字符的大小写是敏感的，但是在Windows中就不是这样。而当你装载一个Windows系统中的驱动器后，对其上的文件操作时，字符的大小写就变得不敏感了，这意味着用户不可以在装载的Windows驱动器的同一目录中建立这么两个文件：ABC.TXT和abc.txt，因为在Windows系统中，它们代表同一个文件。

（二）装载实例

1. 装载软盘

首先在/mnt目录下为软磁盘创建一个目录floppy（有时Linux系统在安装的时候已经为你做了这一步）：

mkdir /mnt/floppy

接着用装载指令将软盘中的内容装载到这一目录中：

mount —t msdos /dev/fd0 /mnt/floppy

此后你就可以在/mnt/floppy下完全访问到软磁盘中的内容了。当然你亦可以用vfat这一文件系统类型代替msdos以使你能正确访问到软盘上的长文件名字，或者是用ext2代替它以使你可以访问到Linux文件格式的软盘。

2. 装载FAT 32格式的C盘

在/mnt目录下为之创建一个目录winc（winc以及其他的装载目录是笔者的假设，你可以用自己喜欢的其他合法字符去命名这些目录）：

mkdir /mnt/winc

用装载指令将C盘内容装载入该目录：

mount —t vfat /dev/hda1 /mnt/winc

3. 装载光盘

在/mnt目录下为你的光盘驱动器创建一个目录cdrom（有的Linux系统安装时已为你完成了这件事）：

mkdir /mnt/cdrom

如果你的光盘驱动器安装在primary slave上，设备文件名就是/dev/hdb；如果安装在secondary master上，设备文件名就是/dev/hdc。假设你的光盘驱动器挂在secondary master，就用下面的装载指令：

mount —t iso9600 /dev/hdc /mnt/cdrom

由于Linux版本的不同，你所用的系统或者会使用这样的指令来装载光盘：

mount /dev/cdrom

或mount /mnt/cdrom

二、卸载（umount）

如果你已经用指令将软盘装载进相应目录，就不要直接将它们从软驱中取出来，否则可能会导致信息丢失；已经装载的光盘驱动器更是会令面板上的弹出键暂时失效以阻止你将光盘直接取出。

在你取出它们之前，首先要确定已经没有用户对它们进行访问操作，包括没有工作窗口处在该磁盘驱动器被装载的目录。卸载指令的格式如下：

umount 卸载目录名

如用户想要卸载软盘，可以用如下指令：

umount /mnt/floppy

三、进阶使用

（一）为装载指令增加简捷的形式

用户可以在系统的配置文件/etc/fstab中指定一些常用的需要装载的驱动器，以使用更简捷的指令来装载它们，下面给出一个添加Windows系统的C盘进配置文件的例子：用文件编辑工具vi（也可以用其它你熟悉的文件编辑工具）打开/etc/fstab，我们会看到系统已经为根目录“/”、软磁盘驱动器、光盘驱动器等指定了文件系统类型和设备文件名以及装载目录（可能会因为所用的Linux版本不同，被指定的驱动器会有出入，不过用户可以参照下面的例子做出改动）。你既可以改变它们的预设值，也可以在其中添加新的驱动器。用箭头将光标移到最后一行，将下面一行加入文件：

/dev/hda1 /mnt/winc vfat defaults 0 0

如果还想装载硬盘内的其它分区，可以依次加入。完成后保存文件并退出。不要忘记相应的创建目录/mnt/winc。此后你就可以用如下指令装载Windows系统的C盘了：

mount /mnt/winc

（二）启动时自动装载磁盘驱动器

用户可能会希望系统在启动时可以将你惯用的磁盘驱动器（如硬盘中的Windows分区或者是光盘驱动器）自动装载，以减轻每次启动后都要手动装载的麻烦。你可以用以下方法实现自动装载的功能：

用文件编辑工具打开/etc/fstab，将刚才我们加入的那一行中的defaults改成auto，如：

/dev/hda1 /mnt/winc vfat auto 0 0

对于其它想在启动时就装载的磁盘驱动器，都可以将defaults改成auto。需要注意的是，对于硬盘上的Linux分区（包括ext2和swap）并不需要作出这样的改动。

重新启动系统以后，在/mnt/winc目录下，你就可以看到系统在启动时就装载好的Windows系统的C盘了。

四、使用Mtools

在Linux中想要共享Windows系统资源，除了上面介绍的装载（mount）命令之外，还有一组名为Mtools的软件包可以实现这一功能。和大多数实用工具一样，Mtools软件包亦是Linux系统默认的安装套件。你可以在http://mtools.ltnb.lu/

（一）Mtools命令

对于熟悉DOS命令的用户来说，Mtools命令会让他们感觉到像回到了DOS的世界。在下面的表格中我们可以看出它们有多么相似：

很显然，Mtools命令是简单地在DOS命令前加上了一个m前缀，功能还是和在DOS下一样的。须要注意的是，更改目录路径在DOS命令中可以有cd和chdir两种命令格式，但是在Mtools中只有mcd一种格式。类似的在Mtools中没有对应的命令还有mkdir、rmdir、rename等等。

（二）mtools.conf文档的配置

因为Mtools是一个仿真DOS命令的软件包，所以它也保存着DOS系统下的磁盘驱动器概念。Mtools不但可以实现A盘 、B盘、 C盘的概念，对于一些特殊驱动器（如ZIP驱动器）更有专门的命令（mzip）以让用户方便地进行操作。在默认的环境下，A盘和B盘分别对应着两个软盘驱动器（如果有的话），N盘则对应着DOSMU启动盘的镜像文件。

我们可以通过修改/etc/mtools.conf文档来改变默认的配置，当然在做出修改之前你仍要对Linux下的设备文件名有相当的认识才行（请参考上一期本版有关设备及其文件名的介绍）。假设你的机器有一个1.44MB软驱（A），硬盘挂在第一个IDE接口的主位置上，在第一个分区（C盘）中安装有Windows操作系统，光驱挂在第二个IDE接口的主位置上（D盘），以及SCSI盘等，则可以对mtools.conf文档简单配置如下：

mtools.conf

drive a: file=″/dev/fd0″ exclusive 1.44m

drive c: file=″/dev/hda1″

drive d: file=″/dev/hdc″

drive X: file=″/dev/rdsk/c0t5d0s2″ partition=4 scsi=1 nodelay

（三）Mtools命令的使用

Mtools的命令系统和DOS有很多相似之处，比如在命令后面可以加不同的参数以实现更为丰富的功能（Mtools命令的参数使用方法是在命令后面加“－<参数>”，你可以把它想象成用“－”代替了DOS命令使用的“/”），可以使用通配符“?”和“*”。更为方便的是，使用Mtools不需要事先装载（mount）和事后卸载（umount）。

下面我们以目录列表命令mdir为例作说明(想知道其它命令的使用格式和更多的可用参数可以用man命令查看相关帮助系统)。mdir的功能是对DOS目录和其中的文档进行列表操作。命令格式如下：

mdir [－/] [－f] [－w] [－a] [－X] msdosfile [msdosfile]

各命令参数的功能分述如下：

/：输出当前路径下的所有目录和文档，相当于DOS命令dir中的“s”参数；

f：列表时不尝试计算当前分区的自由空间，对于大硬盘来说，这样做可以节省一些读入和扫描分区表（FAT）的时间；

w：列表时在一行中显示多个目录文档，这种输出格式将不显示文档的大小和创建时间，相当于DOS命令dir中的“w”参数；

a：列出隐藏的目录文档；

X：简要列表， 列出路径名而不列出其它附加的信息。

一些具体的应用实例：

1. 列表A盘下的所有目录文档（不包括隐藏的目录文档）

mdir －/ a:

2. 列表C盘Windows目录下含有“abc”的目录文档

mdir c:/windows/＊abc＊

或者mdir c:\windows\＊Abc＊

注：在Mtools命令中，/和\是可以混用的。又因为列表的是DOS系统下的文档，对大小写并不敏感，所以“abc”和“Abc”在这里是等价的。而通配符“＊”的用法和DOS命令亦有所不同。

五、在Windows系统中共享Linux系统的文档

无论对于刚接触Linux的新手还是对之已经熟悉的高手，恐怕都免不了想在Windows系统中调阅Linux系统的文档。这可以使用在Windows系统下观看Linux分区的小软件（如fsdext2等）。这里再向各位推荐两种可以实现这种功能的小软件。

(一)Linuxindos

Linuxindos是一个不到500KB的小程序，笔者找到的版本是Beta 0.9的DEMO版，用户可以在http://best.163.com/

在Windows系统下将linuxindos.zip解压到硬盘目录，用户就会看到一个带着企鹅图标的可执行程序LiD95Demo.exe。启动程序后，会出现一个接口非常简单的窗口，当用户挑选一个含有Linux系统文档的磁盘驱动器以后，Linux分区中的所有文档就会显示出来，左边窗口列出的是Linux分区中的目录，而右窗口就列出了目录内的文档，感觉就像在Windows的资源管理器中一样。

（二）Explore2fs

这是一个更为小巧的自由软件，大小约308KB，用户也可以在http://best.163.com/

在右边视窗中选择好文档后，按下鼠标右键可以选择相应的操作，“Export file”是直接导出文档，而“Export to Text”则是除去文档的分隔符后导出和TEXT兼容的文本文档。
<hr>

做张grub启动盘

有时候我们的电脑可能出现系统无法引导的情况，比如说你重装了windows，把MBR给重写了，又比如像我今天这样，把装了grub的系统给格了，电脑启动后无法读到引导信息出错，什么系统都进不了了。
这时你当然可以用安装时或什么管你什么时候做的启动盘启动LINUX(然后再进行修复)，用windows启动盘启动windows。
但这样你得至少做两张启动盘啊（呵呵，当然用loadlin之类的也行，不过我觉得更麻烦）。而且你每装一个LINUX就得做一张启动盘，不是很麻烦吗？
其实没有那么麻烦，只要你用grub做的引导，就很容易了，那么就是做一张grub启动盘.
这样你就相当于把grub装到了软盘上，利用一些grub命令就可以做到你的MBR没有被破坏前能做的所有事。而且，你在A机器上做的启动盘还可以拿到B、C、D……等等其它机器上用。
制作方法：

首先你用的引导程序必须是grub。
确认了这一点之后进入/boot/grub目录：
cd /boot/grub
然后把stage1和stage2两个文件写到你的软盘上去（呵呵，当然别忘了把软盘放进去哦）：
dd if=stage1 of=/dev/fd0 bs=512 count=1
dd if=stage2 of=/dev/fd0 bs=512 seek=1

这样一张grub引导盘就做好了，下面就谈谈使用方法（虽然很简单，但还是怕有人不清楚）

首先用这张启动盘启动后会出现一些关于grub的信息，然后就是如下：
grub>
这就是在等你输入grub命令来启动系统。
对于LINUX，一般需要如下三个命令：
root,kernel,boot.
ROOT命令就是让你告诉GRUB，你的LINUX系统装在哪个分区，KERNEL命令就是让你指出用哪个内核启动，BOOT当然就是开始引导啦。
举个例子：
grub>root (hd0,8)
File system Type is ext2fs.(这一行告诉你文件系统是什么，详细的显示内容可能跟我写的不一样)
grub>kernel /boot/vmlinuz ro root=/dev/hda9
grub>boot

这样你的LINUX系统就启动了，注意的一点是GRUB中关于分区的叫法跟LINUX有点不一样，比如上面的(hd0,8)就表示hda9.是的，hdX就代表第X＋1个硬盘，hd0就代表第1个硬盘（相当于hda）,8就代表第9个分区（GRUB中的表示方法跟C语言有点像，下标是从0开始的），(hd0,8)当然就是代表hda9啦。
知道这些应该知道怎么引导LINUX系统了吧？注意一点，如果你不清楚具体的内核文件名（比如有的可能是vmlinuz-2.4.20什么的），那么你可以用TAB键自动补全，相当方便，你只要打入kernel /boot/vm然后再按一下TAB键，那么就会显示全部在/boot/下以vm开头的文件。

引导其它系统就方便多了（比如windows,FreeBSD等），你要做的也是三个命令，ROOT跟BOOT命令跟LINUX一样，反正就是用ROOT命令指定你装的这个系统所在的分区，BOOT命令就是开始引导系统。不同的是把引导linux的kernel命令改成chainloader +1命令。例如：

grub>root (hd0,0)
File System Type Vfat
grub>chainloader +1
grub>boot

呵呵，好了，把你做好的grub启动盘好好保存起来吧，以后装LINUX的时候它如果问你要不要做启动盘时你都可以大胆地说：NO!。因为你的这张grub启动盘基本上可以应付大部分无法启动的情况了。
呵呵，不过这种启动盘无法应付严重的系统错误（比如内核文件受到破坏什么的等等），但是这样的情况我还没有遇见过，我需要启动盘的情况往往是因为引导程序出了问题，比如重装WINDOWS，比如GRUB被我搞掉了等等。